Open-JTAG

LPC2388 ボード ＋ GCC + Eclipse 版

マニュアル

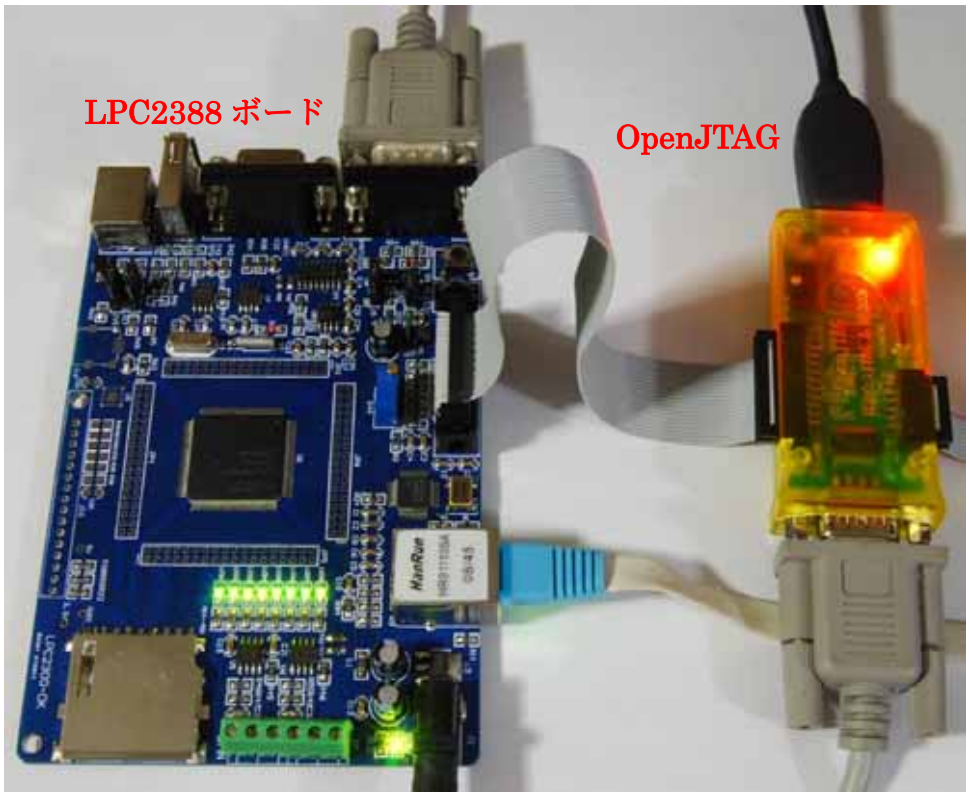
株式会社日昇テクノロジー

http://www.csun.co.jp

info@csun.co.jp

Ver1.4

2009/7/31



※ 使用されたソースコードはhttp://www.csun.co.jp/からダウンロードできます。

第一章 背景

近年，ARM プロセッサが急速に広まっています．さらに、ARM9，ARM11，ARM-M，Cortex といった新たなアーキテクチャが次々と発表されています．これらの新しいプロセッサでは、従来から存在したARMが拡張され、さらにARM9とARM11が追加されました．

本書では、従来から広く利用されているGNU クロス開発ツールを、これらの最新アーキテクチャ/命令セットへ対応させます．最新版のGNU ツールを実際に使って、対応したクロス開発環境を作ります．

組込アプリを開発する際に、2点は非常に重要です。一つは統合開発環境（IDE）となり、もう一つはデバッグ環境です。いろんな統合開発環境ツールがありますが、無料で使用できるオープンソースEclipseプラットフォームはJava アプリを含めて、C/C++アプリを開発用の全世界共通のIDEです、今後、ますます多くの使用者が増えていきます。なお、デバッグツールはオープンソースOpen-On-Chip Debugger(OpenOCD)という便利ツールがあります。

Open-On-Chip Debugger(OpenOCD)は、システムプログラミングにおけるデバッグと埋め込まれた対象装置がないかどうかテストされる境界走査を提供することを目指します。

OpenOCDのGDB と接続するためにOpen JTAG コネクトを使う必要です。ただし、パラレルJTAGの転送スピードが遅いので、本書はカスタマイズ化のOpenJTAGを利用し、Eclipse、OpenOCDなどオープンソースとあわせて、ARMシリーズ開発を行います、例えば、ARM基板からダウンロード、書き込み、デバッグを行います。

GDBは以下の基板をデバッグできます。

① ARM7(ARM7TDMIとARM720t)
② ARM9、(ARM920t、ARM922t、 ARM926ej-s、ARM966e-s)
③ XScale(PXA25x、IXP42x)
④ Cortex-M3(LM3とST STM32)

第二章 ARM シリーズ開発の仕組み・イメージ

組込開発のよいプラットフォームは基本的に4つモジュールを含まれます。

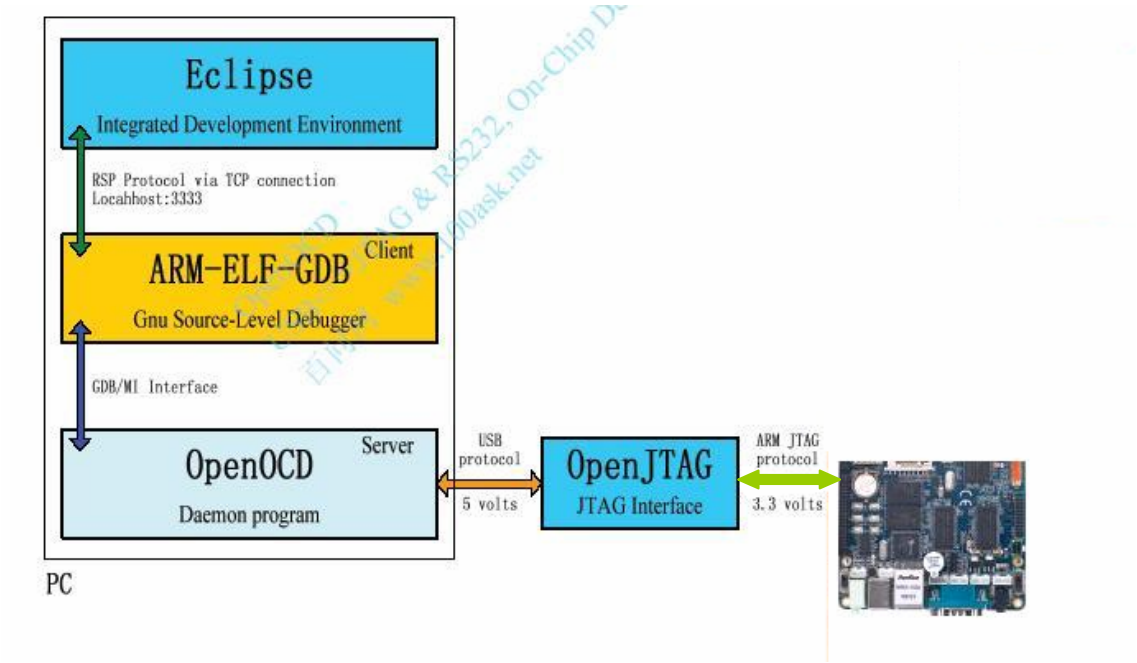
① **統合開発環境：IDE** (Integrated Development Environment）、例えばEclipse

② **クロスコンパイラツールチェイン (Cross Complier tools chain)**

③ リモートデバッグツール（例：OpenOCD、Open On-Chip Debugger）

④ 開発用のパソコンとARM基板のコネクタ（JTAG）

全体のイメージは下記の図です。



第三章 用意するもの

以下は Windows 用です。

ダウンロード URL：

弊社のサーバーから必要ものを一括ダウンロードできます。

http://www.dragonwake.com/download/LPC2388/openJTAG.zip

中身に、下記内容を含まれます。

解凍先：C:¥(お好きな場所を選べられても OK)

ソフトウエアリスト

01.openocd-0.1.0.msi(jtagデバッガ用ソフト)

02.yagarto-bu-2.19.1_gcc-4.3.3-c-c++_nl-1.17.0_gi-6.8.50_20090311.exe(gcc)

03.yagarto-tools-20070303-setup.exe(各種ユーティリティ)

04.jre-6u7-windows-i586-p.exe(Java)

05.eclipse-cpp-galileo-win32.zip(Eclipse)

3.2 OpenJTAG 用のデバイス(jtag デバッガ、弊社で販売しているもの)

①usb-driver(OpenJTAG 用の USB ドライバ)

②interface(OpenJTAG のコンフィグファイルがあります)

③target(LPC2388 のコンフィグファイルがあります)

3.3 LPC2388 用の GCC サンプル：

①LPC2388_LED (動作確認用)

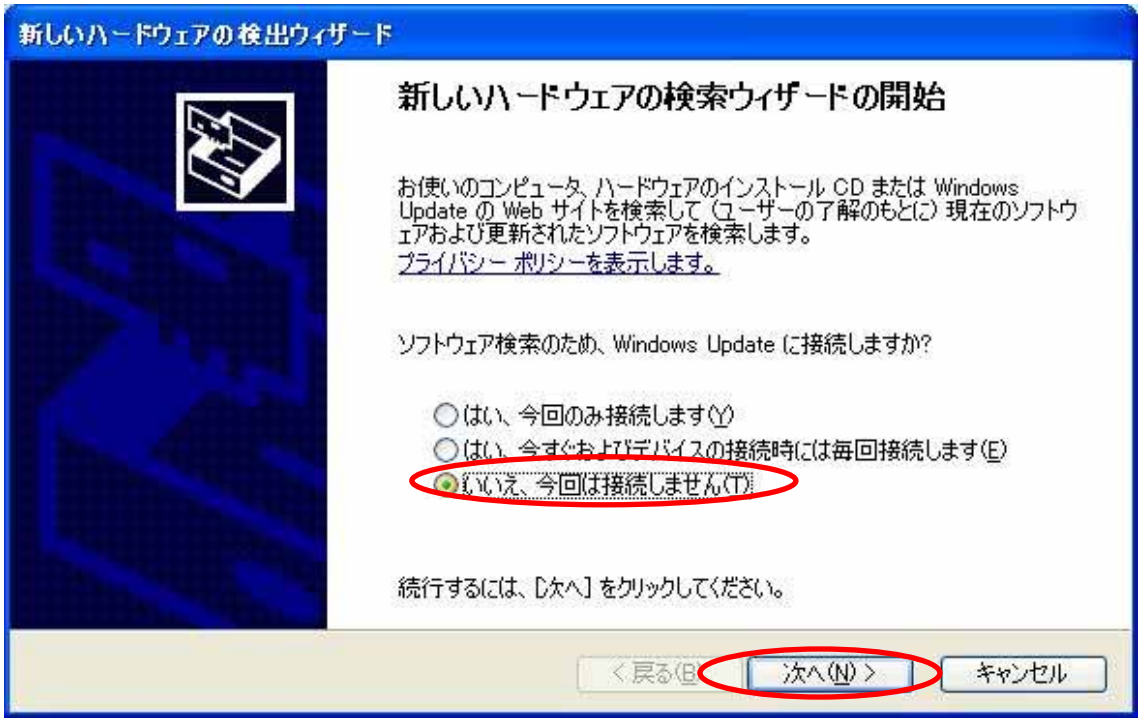
②FreeRTOS(uIP を使った Web サーバ・サンプルです、

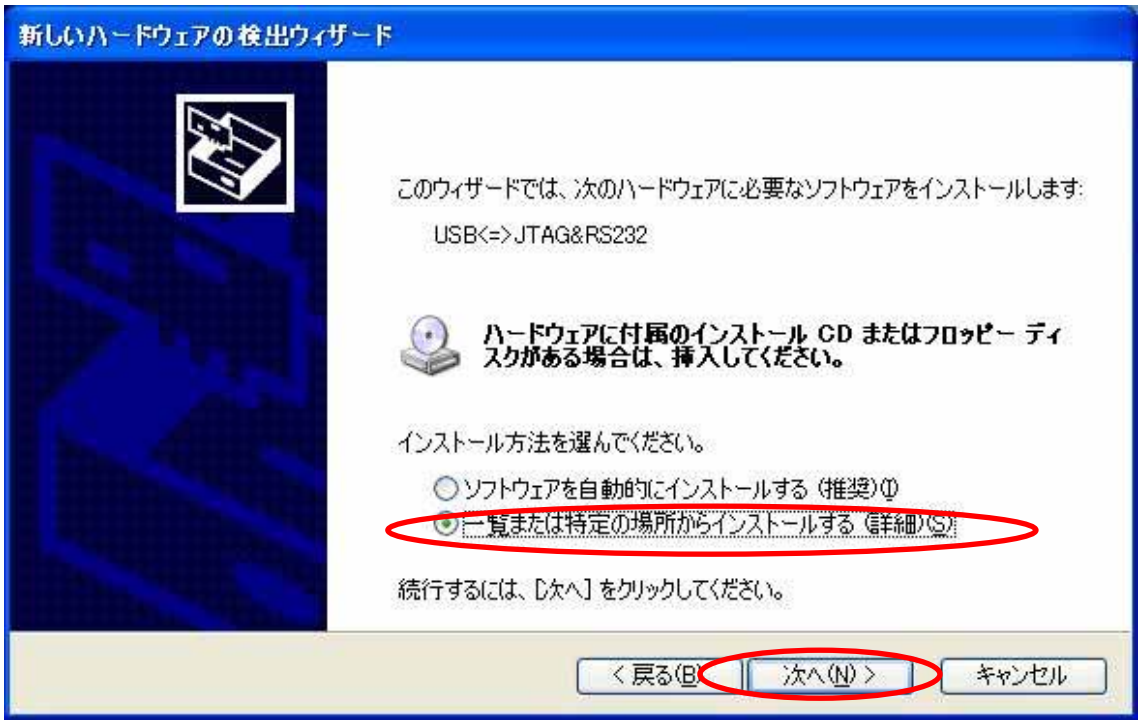
FreeRTOS¥Demo¥ARM7_LPC2368_Eclipse¥readme.txt を参照してください)

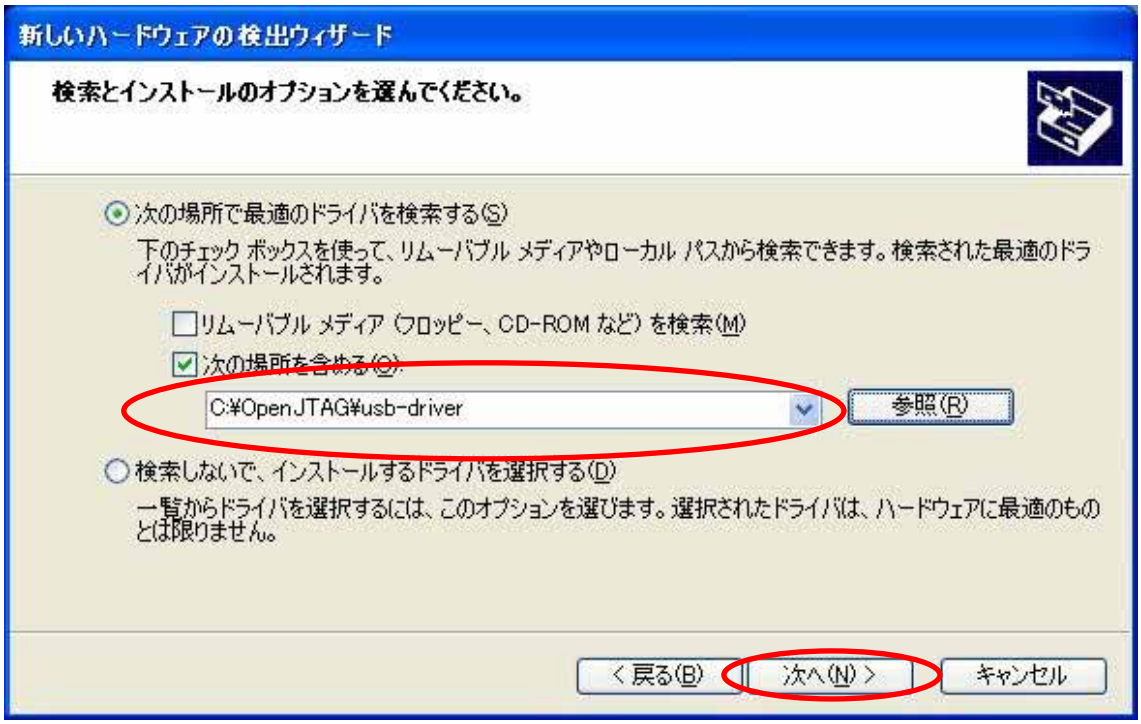
第四章 インストール手順

4.2 OpenJTAG のドライバをインストールする

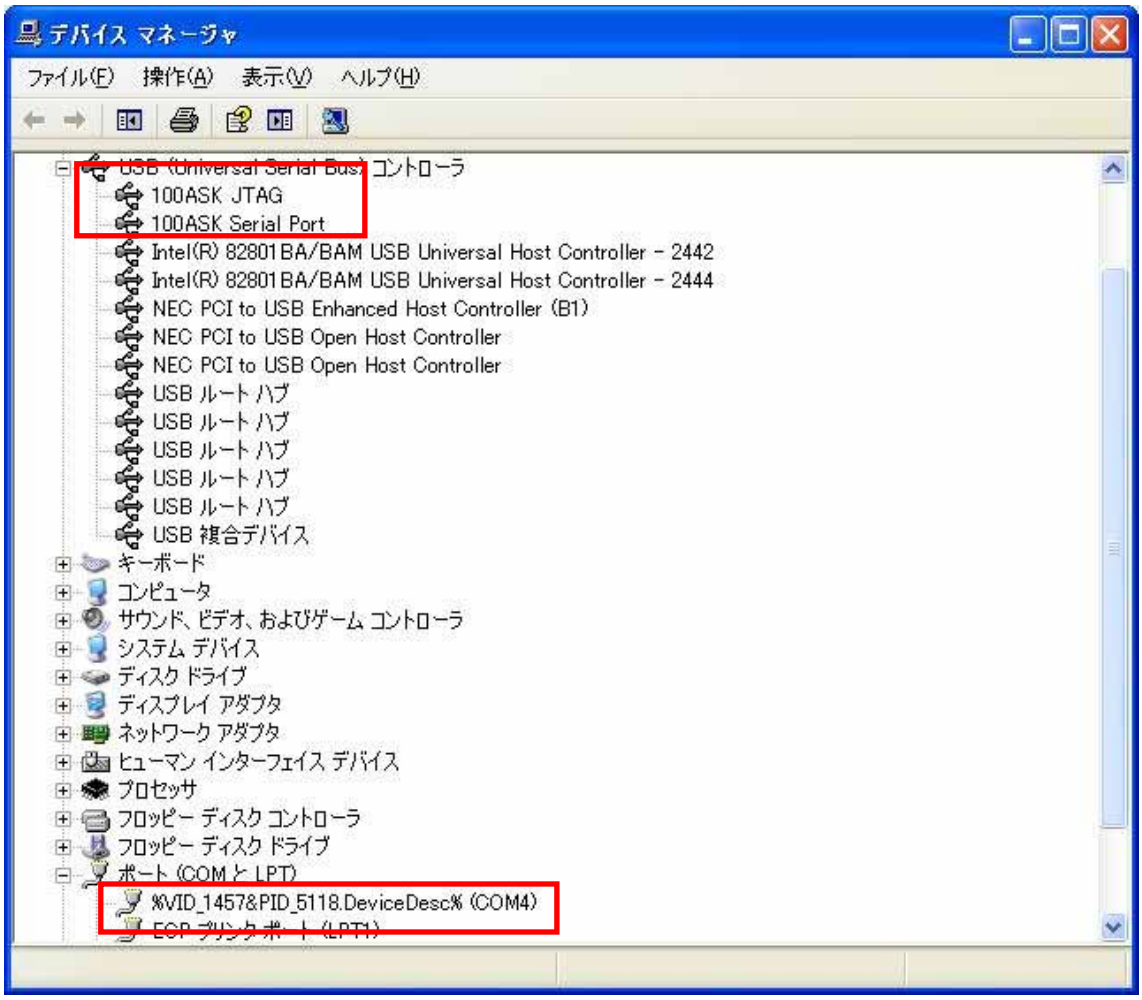
OpenJTAG をパソコンの USB ポートに挿入して、下の通りにドライバをインストールしてください。







USB ドライバのインストールは３回があります。インストール完了すると、デバイスマネージャで三つのデバイスが見えます。



※ OpenJTAG は USB シリアルポートとして使えます。

4.2 ソフトウェアをインストールする

4.2.1　順番で以下のソフトウェアをインストールしてください

01.openocd-0.1.0.msi(jtag デバッガ用ソフト)

02.yagarto-bu-2.19.1_gcc-4.3.3-c-c++_nl-1.17.0_gi-6.8.50_20090311.exe(gcc)

03.yagarto-tools-20070303-setup.exe(各種ユーティリティ)

04.jre-6u7-windows-i586-p.exe(Java)

05.eclipse-cpp-galileo-win32.zip(Eclipse)

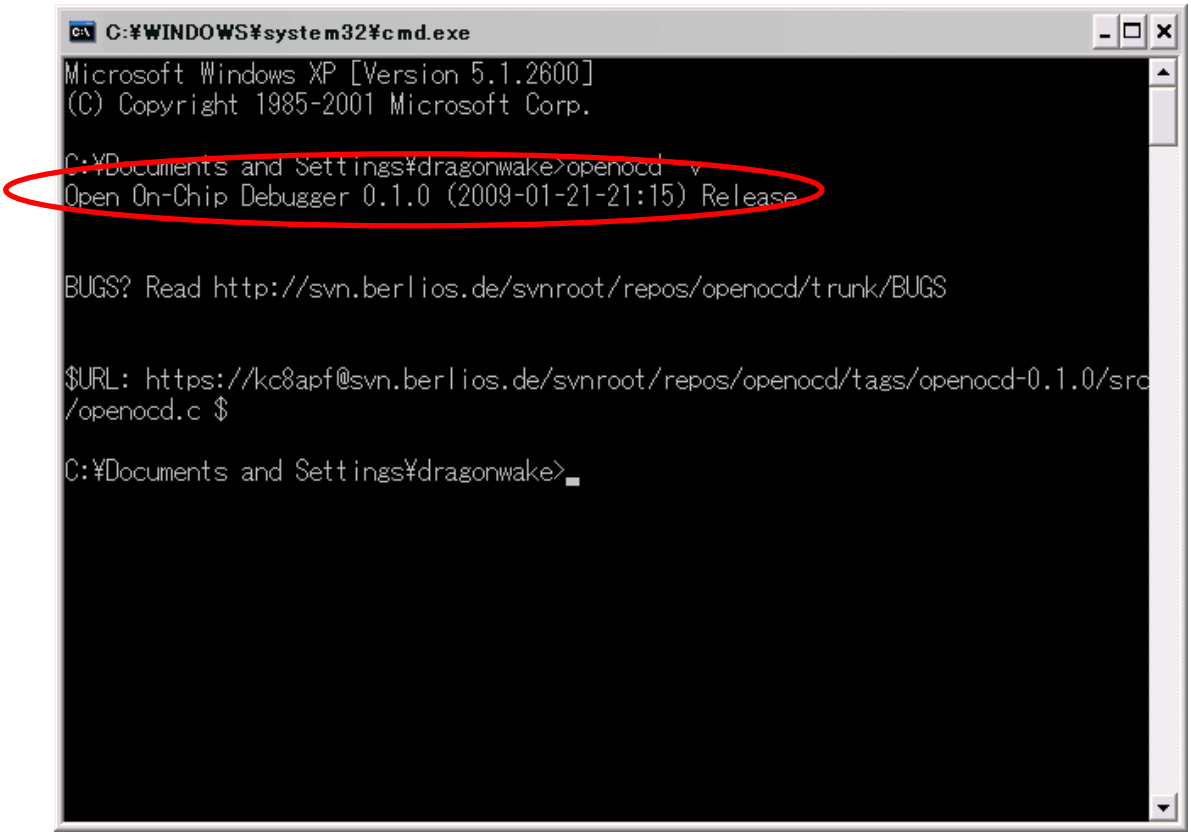
メモ：①もしバージョン 1.4.2 以上の JRE を既にパソコンにインストールされたら、
04 番の JRE のインストールが不要です。
②05 番の eclipse を解凍して OK、インストール不要

4.2.2　ソフトウエアインストール後の動作確認

確認方法：「スタート」→「ファイル名を指定して実行」→「cmd」を入力

①　OpenOCD の確認

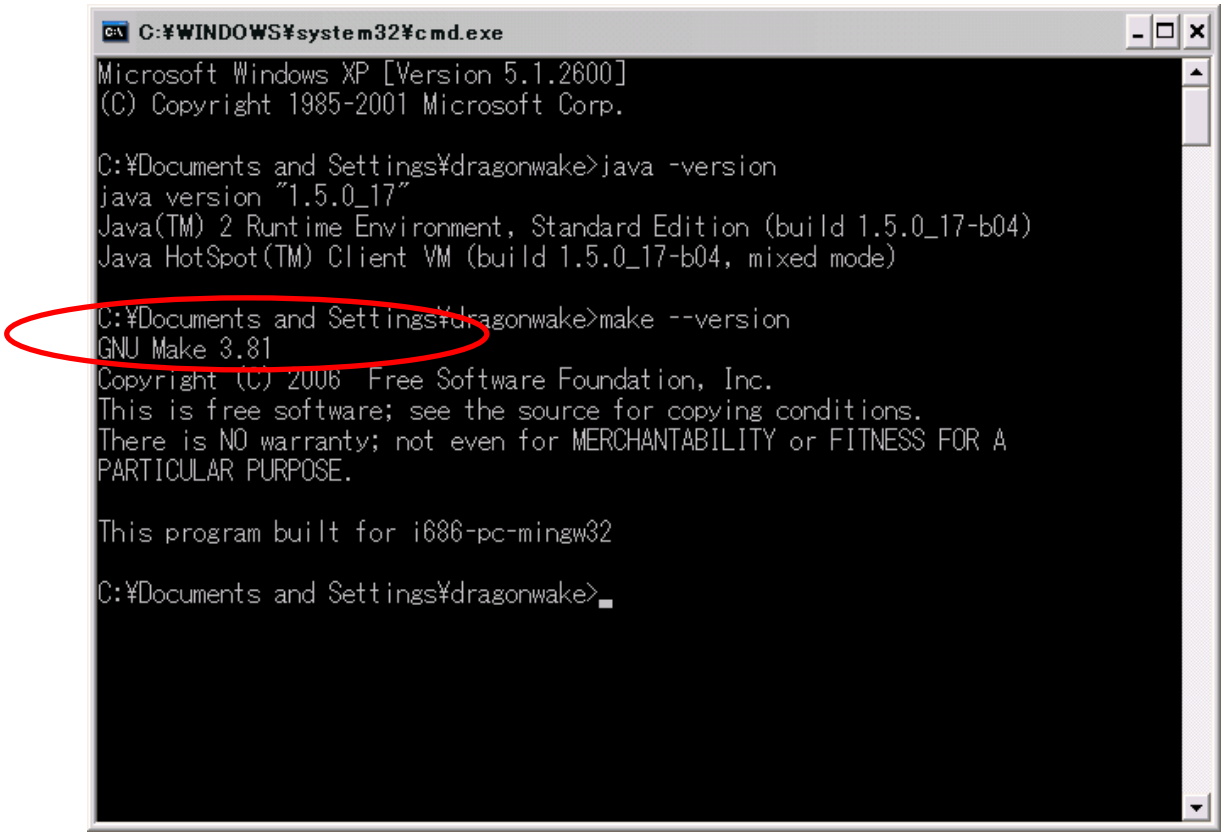
確認コマンド：openocd –v



② コンパイラ確認

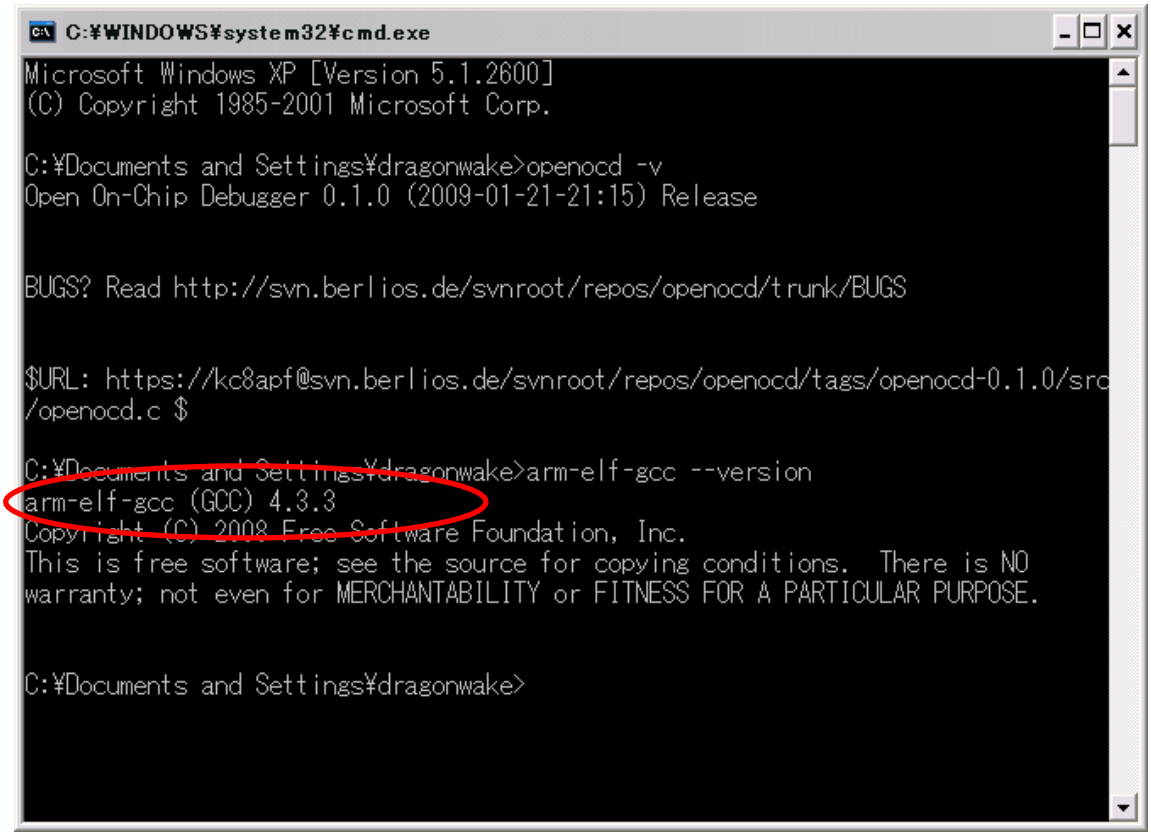
②-1：GCC コンパイラ「make」確認

確認コマンド：make --version



②-1：ARM コンパイラ「arm-elf-gcc」確認

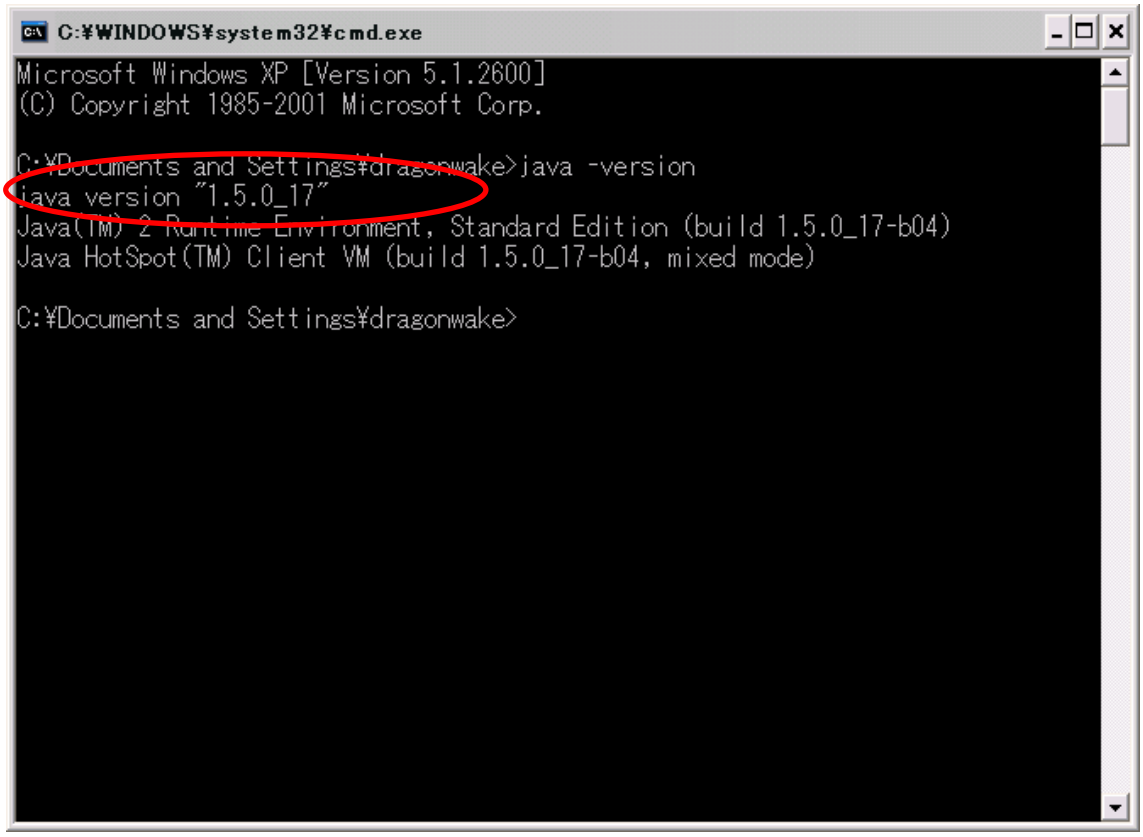
確認コマンド：arm-elf-gcc --version



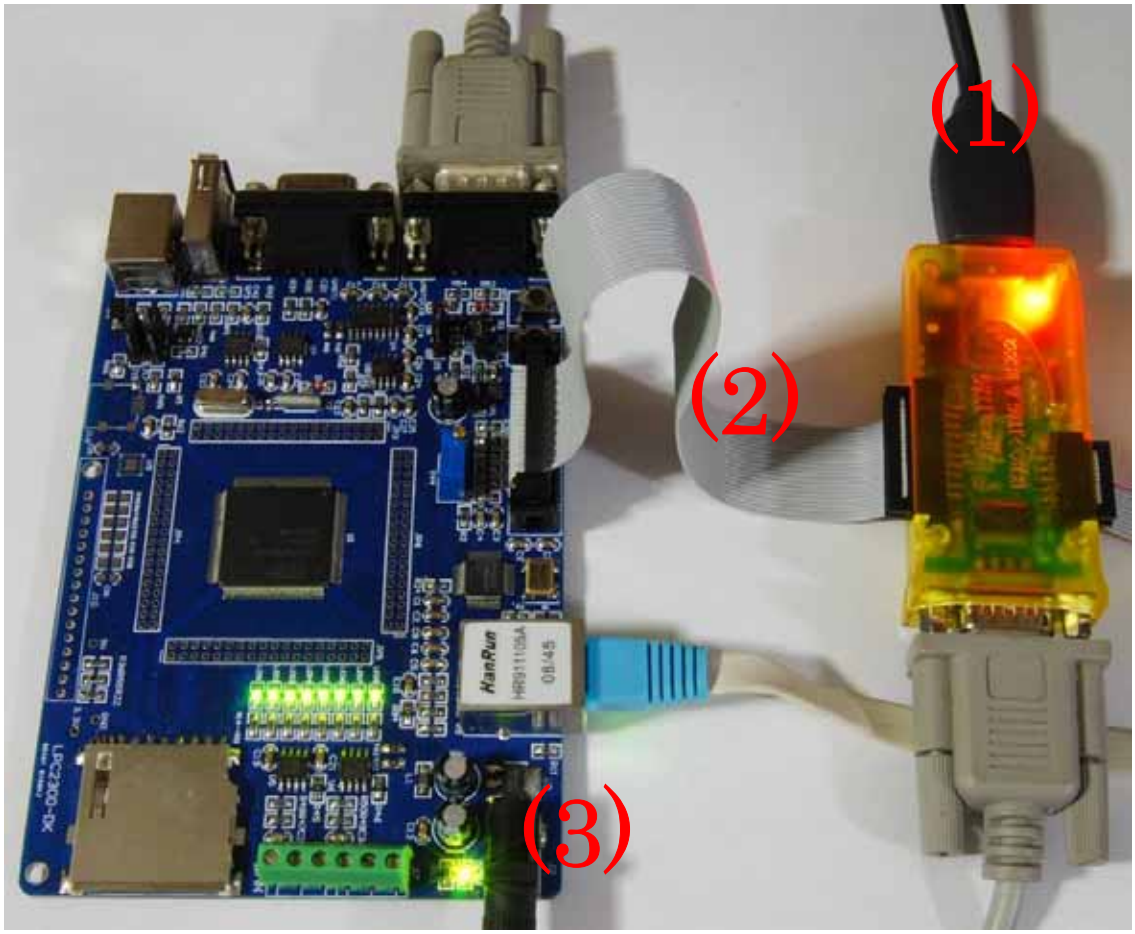
③ JRE バージョン確認：

確認コマンド：

java -version



4.3 動作確認



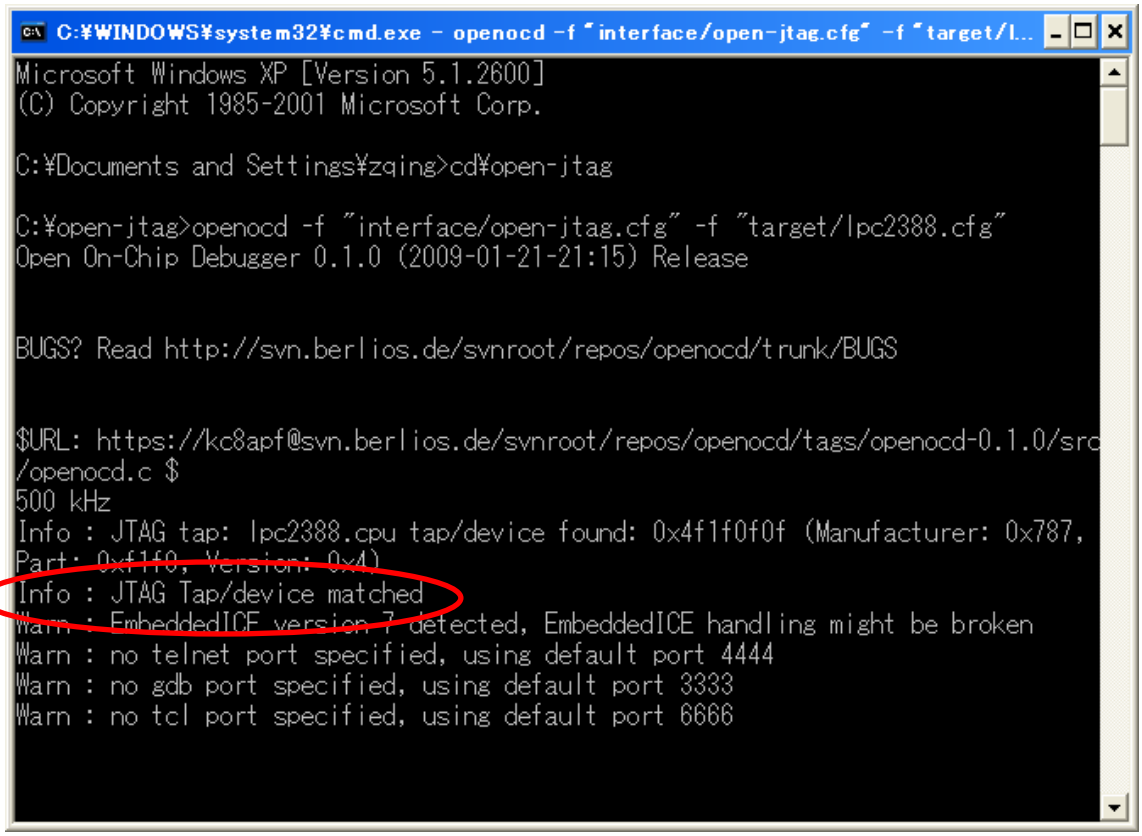
(1). OpenJTAG をパソコンの USB ポートに挿入して、
(2). JTAG ケーブルで OpenJTAG と LPC2388 ボードを繋ぐ
(3). LPC2388 ボードに電源を入れます。

(4). コマンドプロンプトでディレクトリを移動 cd¥OpenJTAG

(5). 下記のコマンドを入力します。

openocd -f "interface/open-jtag.cfg" -f "target/lpc2388.cfg"

※ はじめの-f は OpenJTAG のコンフィグファイルを使います。二番目の-f は lpc2388 のコンフィグファイルを使います。



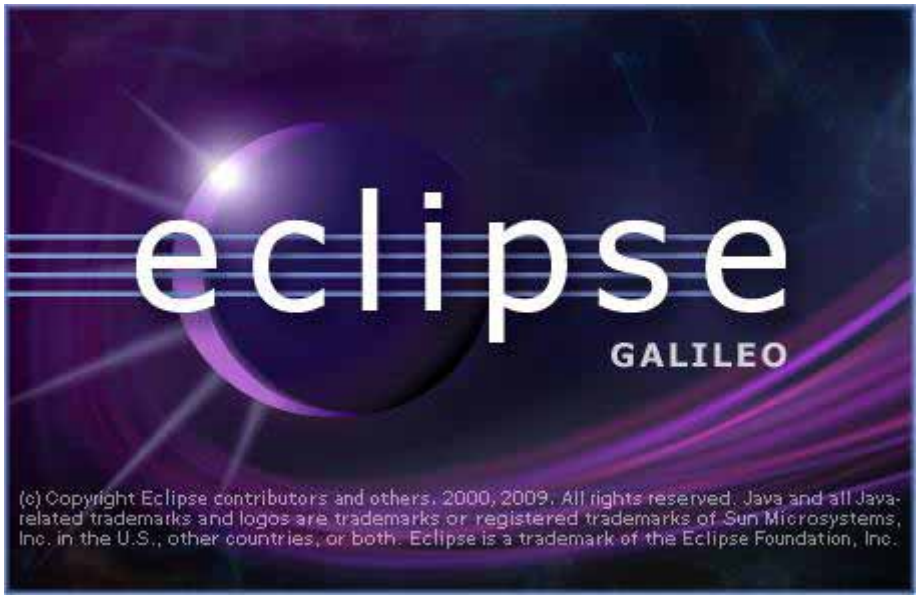
画面のように"Info：JTAG Tap/device matched"と表示されればＯＫです

（この時点で ARM LPC2388 と通信が出来ています）

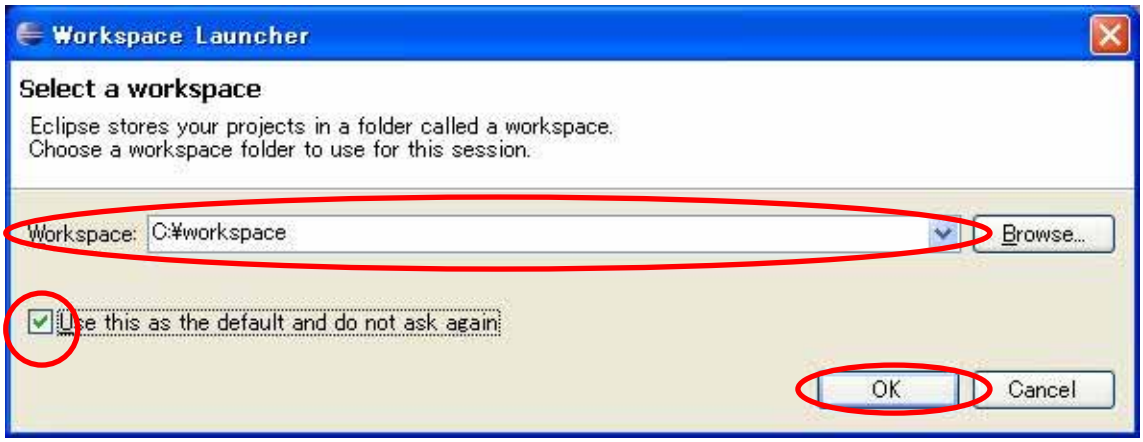
(6). その後 CTRL+C を押してデバッグを中止します。

第五章 Eclipse の設定

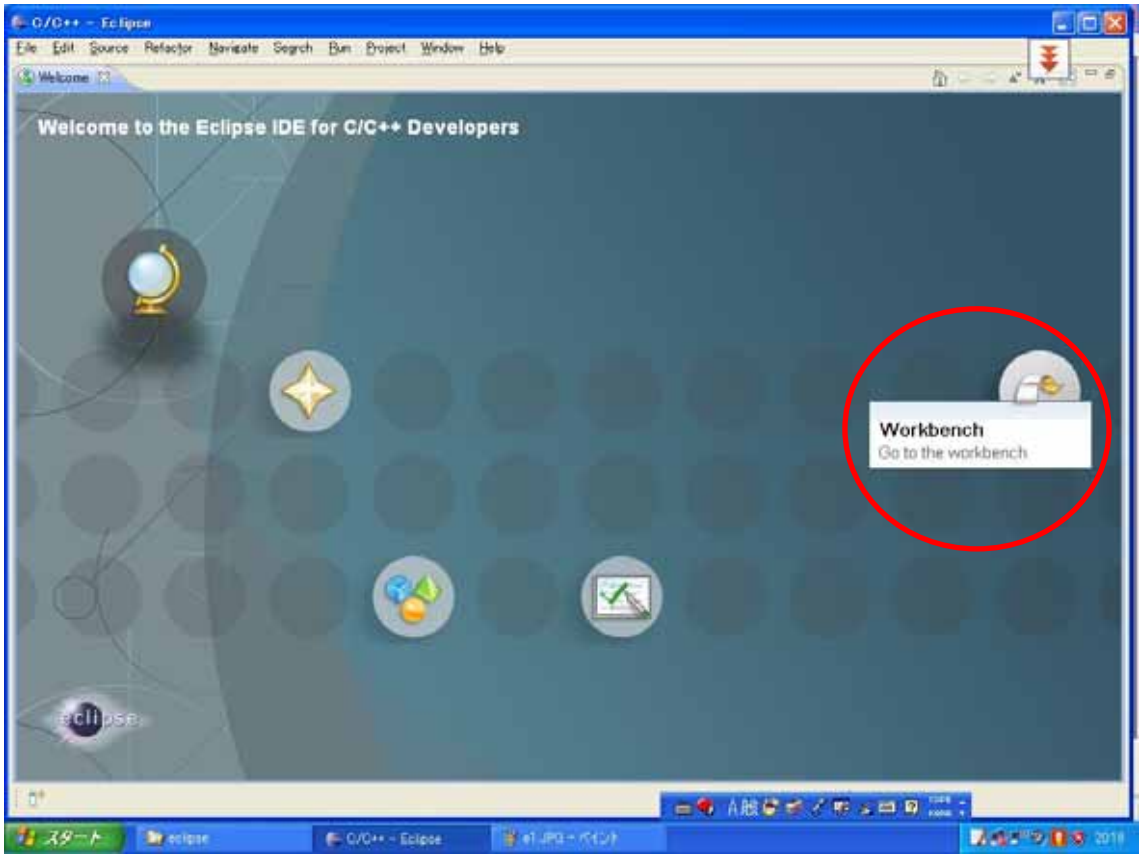
5.1 Eclipse を起動する



最初に Workspace の場所を聞いてきます。デフォルトは名前が長いので、"C:¥ workspace" に変更しました。

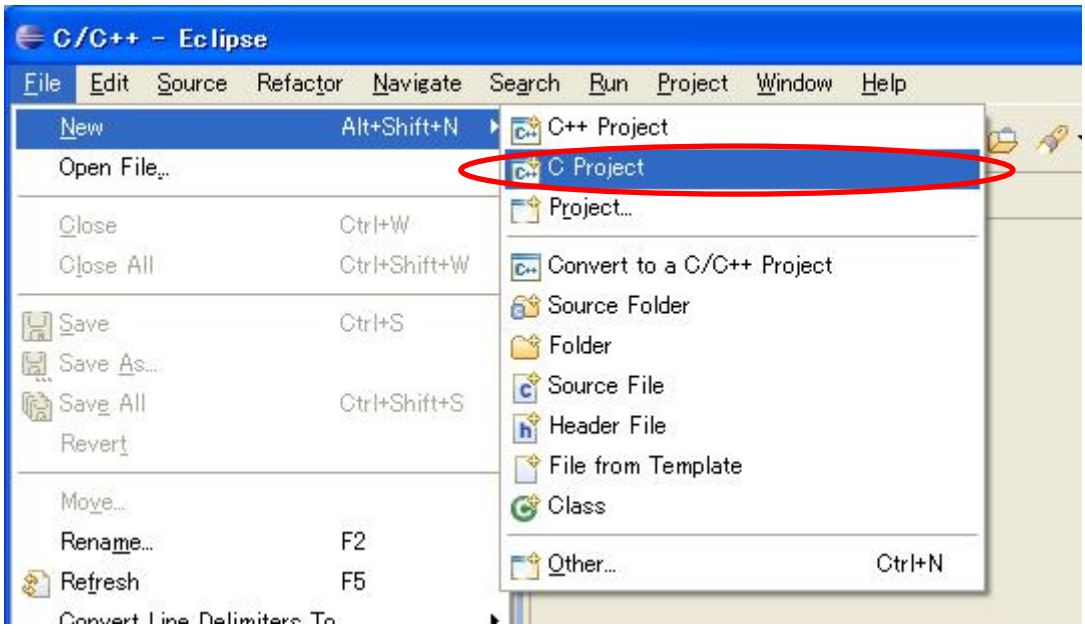


画面の Workbench をクリックします。

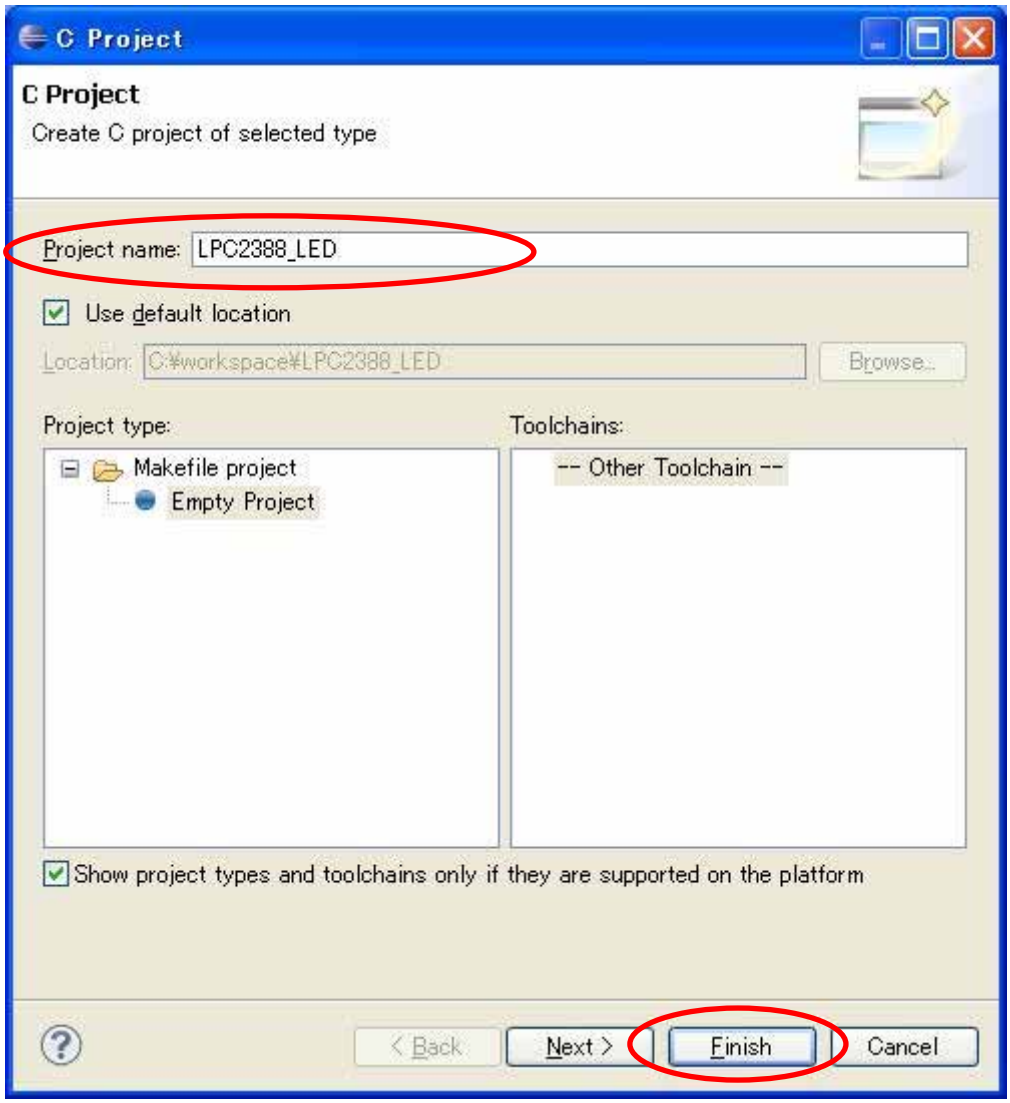


5.2 プロジェクトを作る

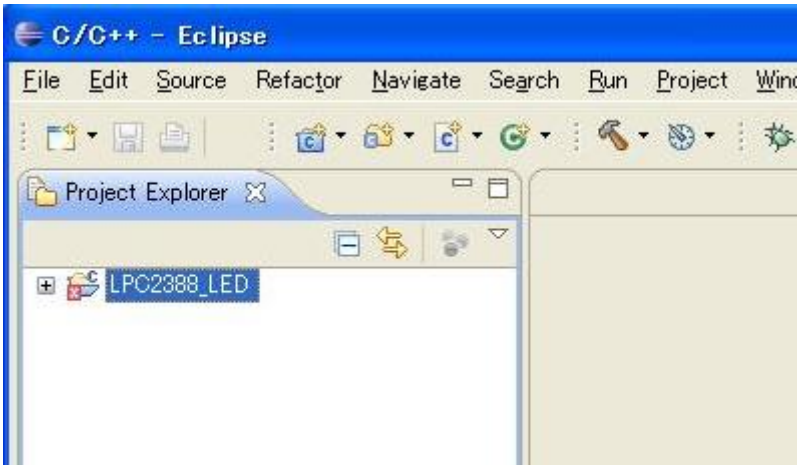
新規プロジェクトを作成するため"File"→"New"→"C Project"を選択します



プロジェクト名を聞かれるので適当な名前(LPC2388_LED)を入力し Finish ボタンを押します。

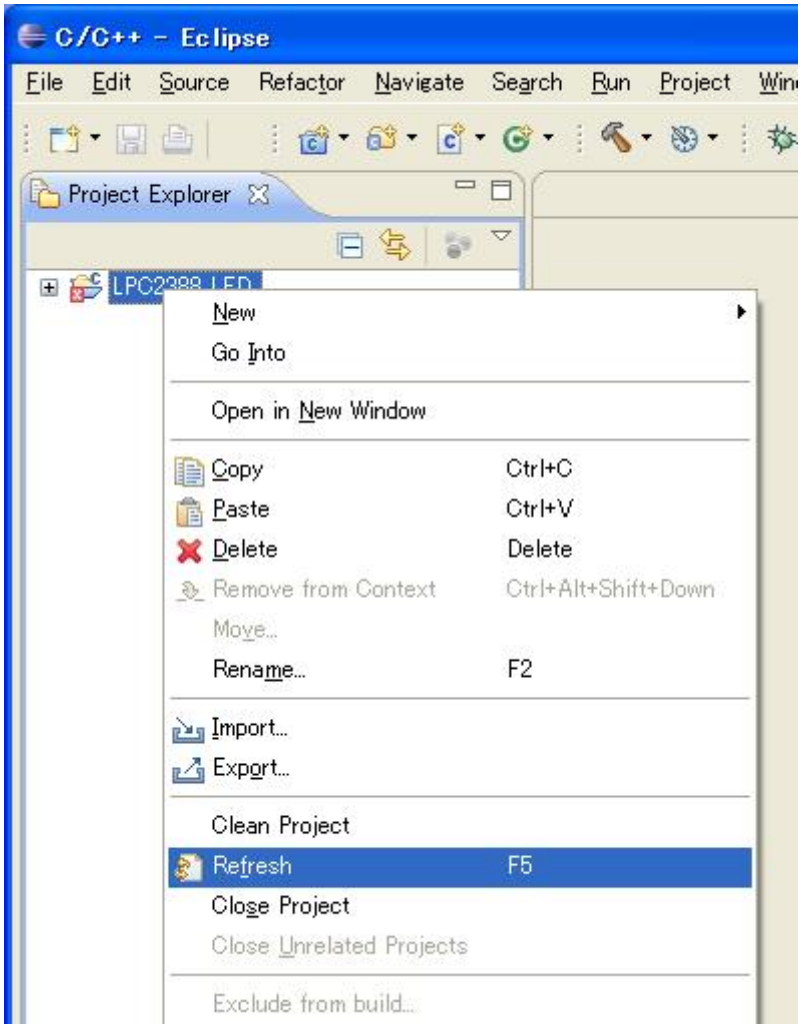


Project Explorer に LPC2388_LED プロジェクトが追加されましたが中身が何もないので、"×"がついています。

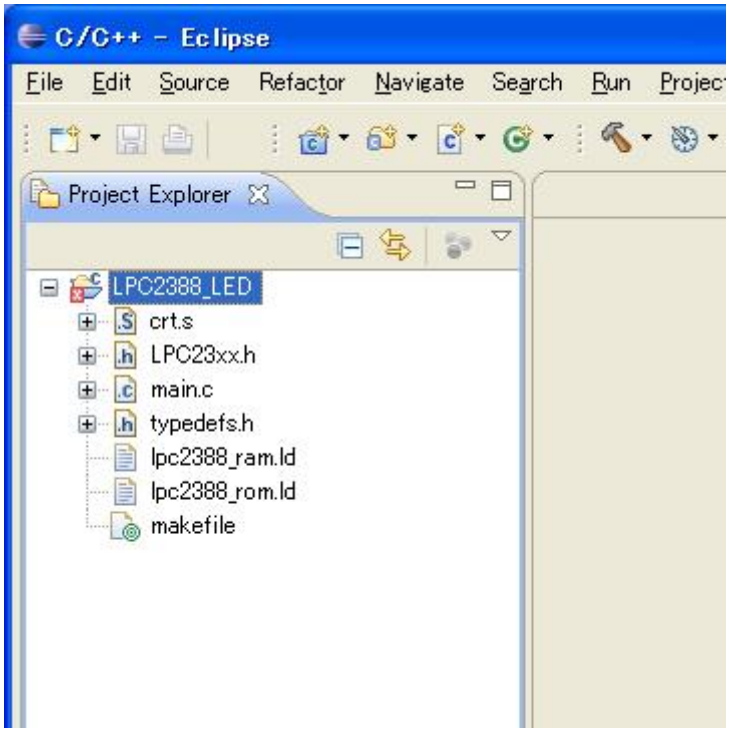


サンプルのフォルダ LPC2388_LED のなかのファイルを"C:¥workspace¥LPC2388_LED" にコピーしてください。

Eclipse の ” File"→"Refresh"を選択します。

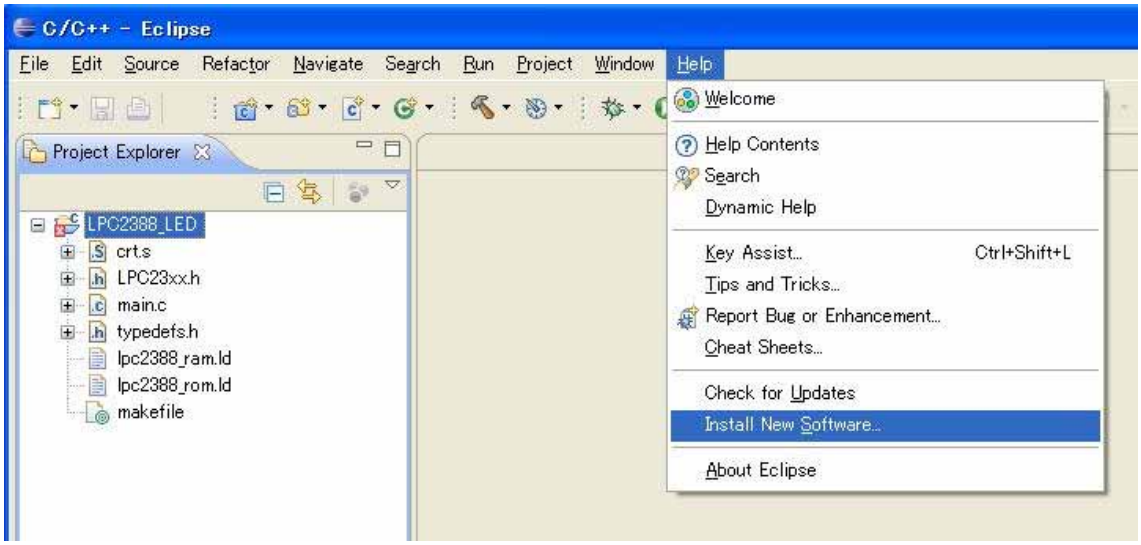


Project Explorer の"LPC2388_LED"プロジェクトの左にある＋をクリックするとファイルの一覧が表示されます。

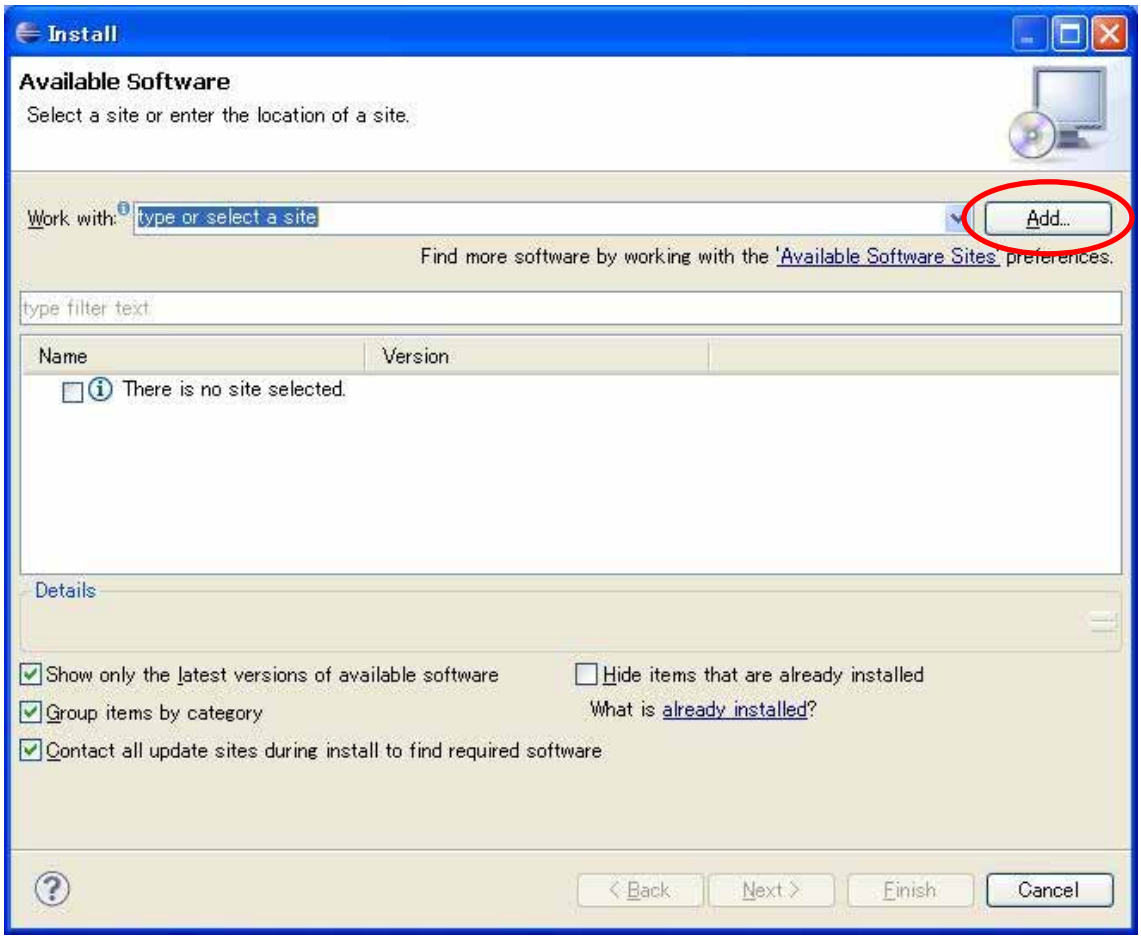


5.3 Eclipse プラグイン(Zylin Embedded CDT)インストール

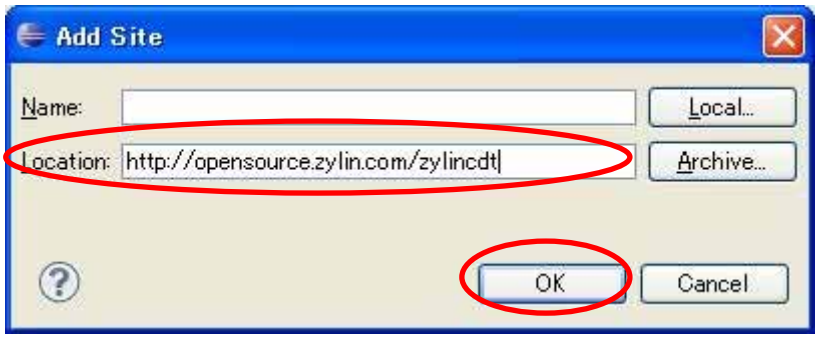
Eclipse の"Help"→"Install New Software"を選択します



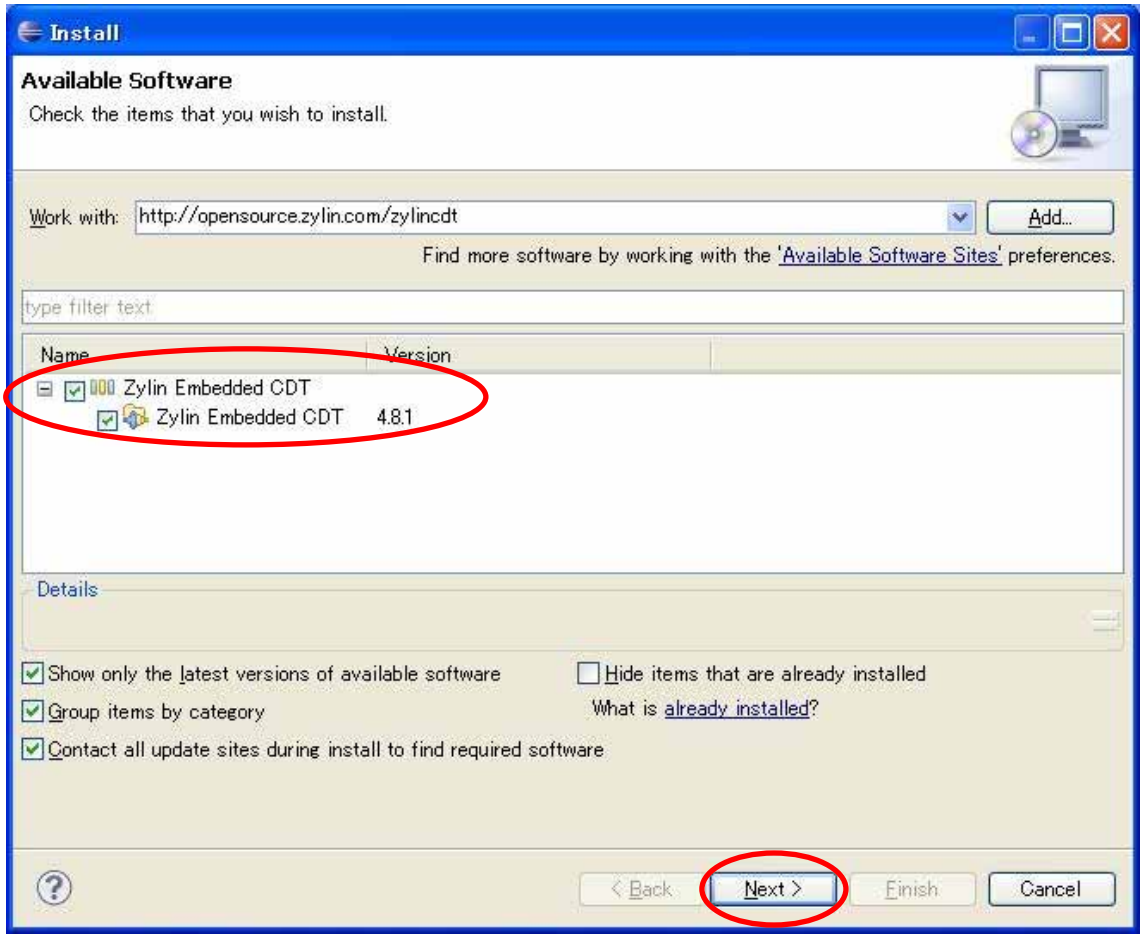
Add ボタンを押します。

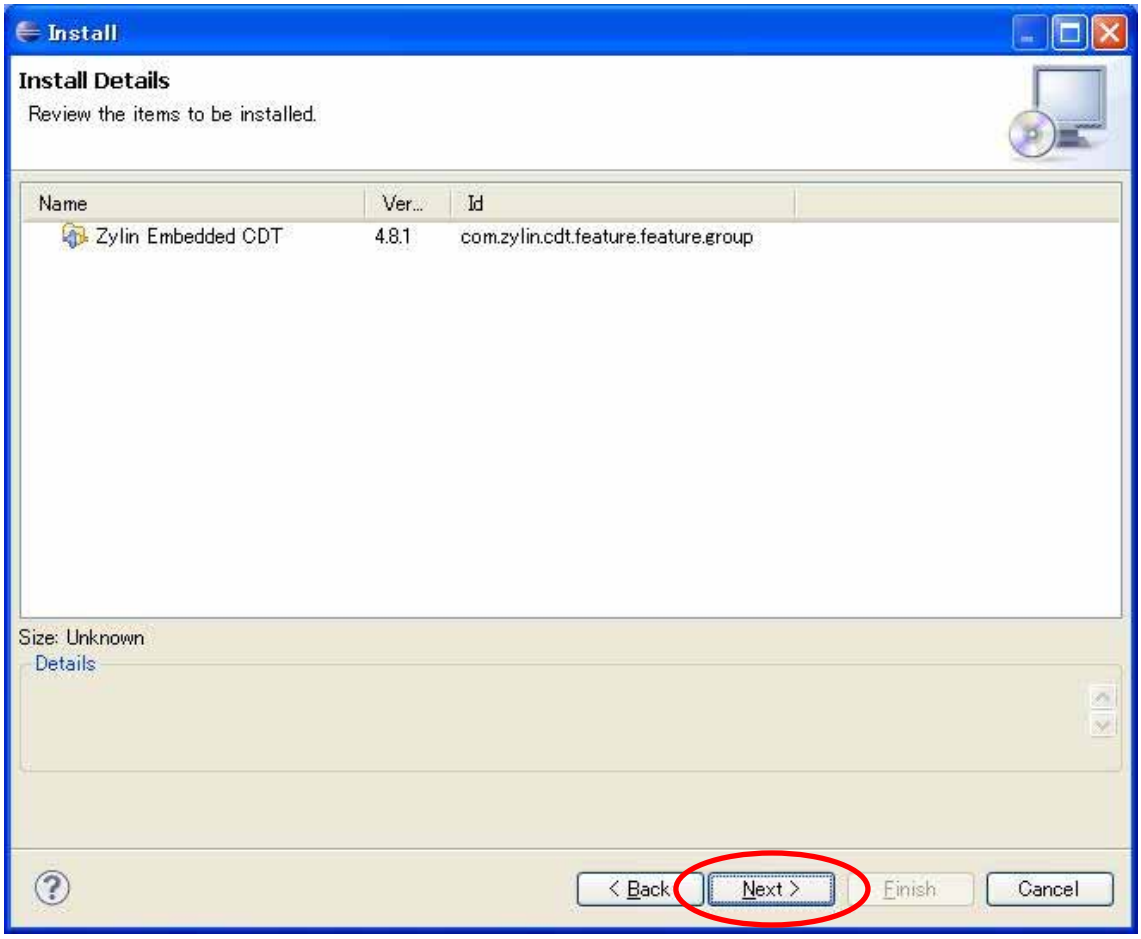


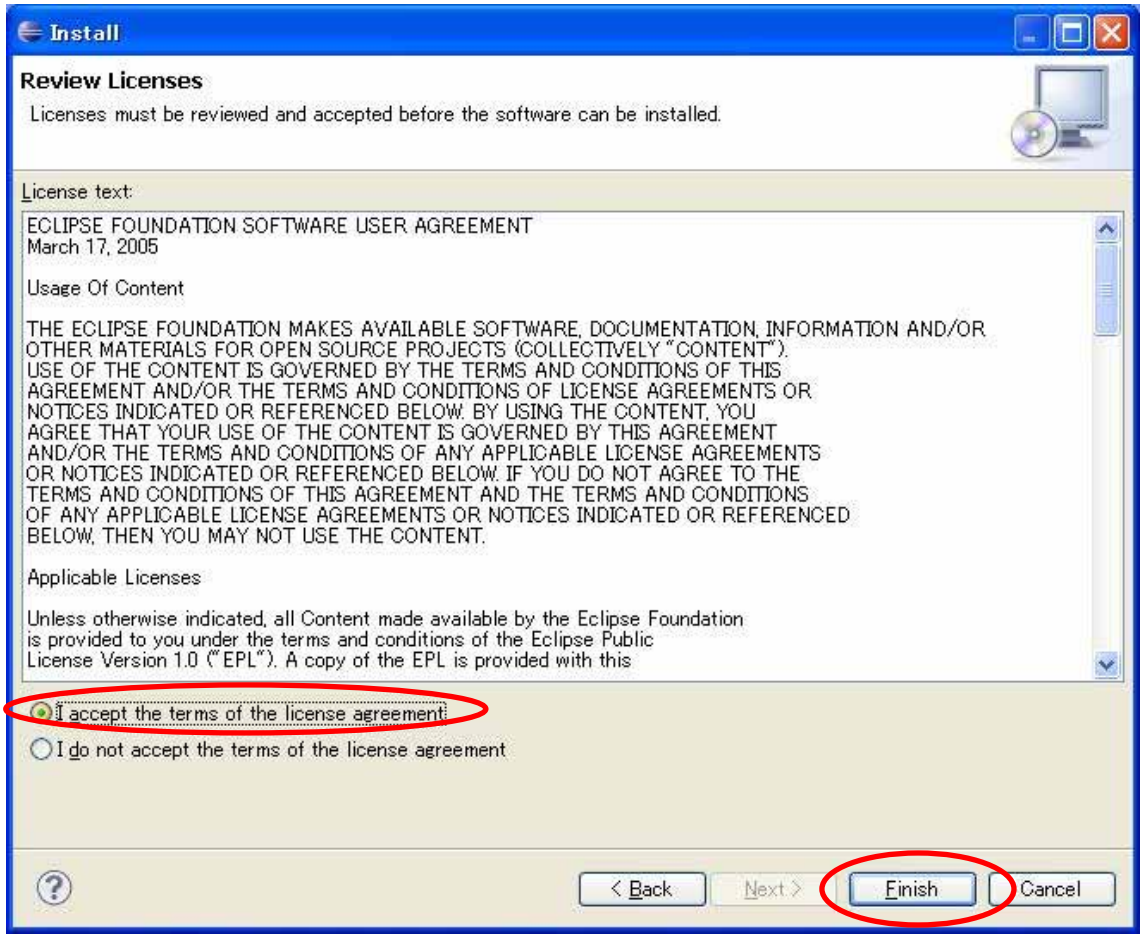
Add Site の"Location"に"http://opensource.zylin.com/zylincdt "と入力し OK ボタンを押す。

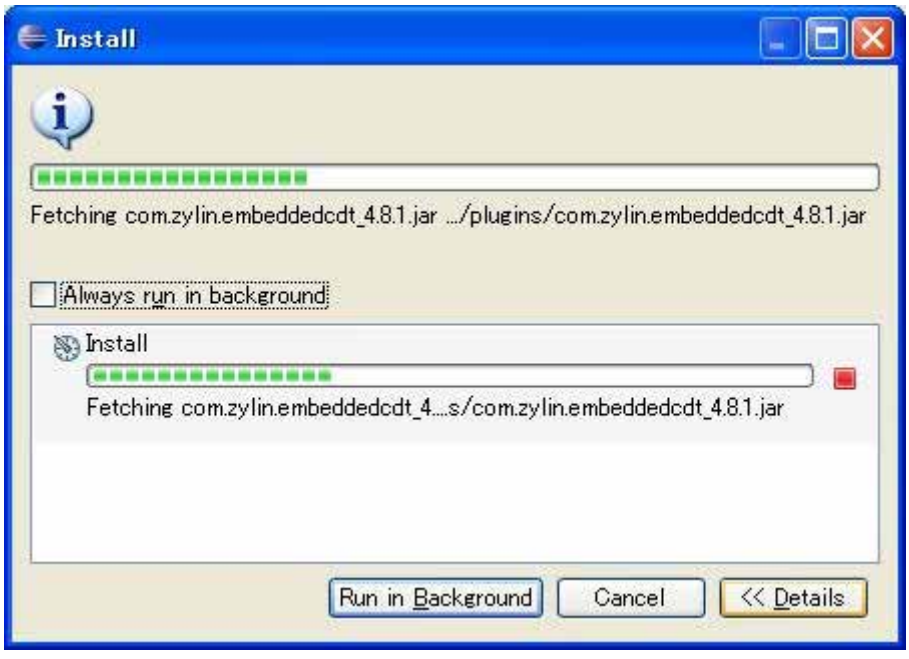


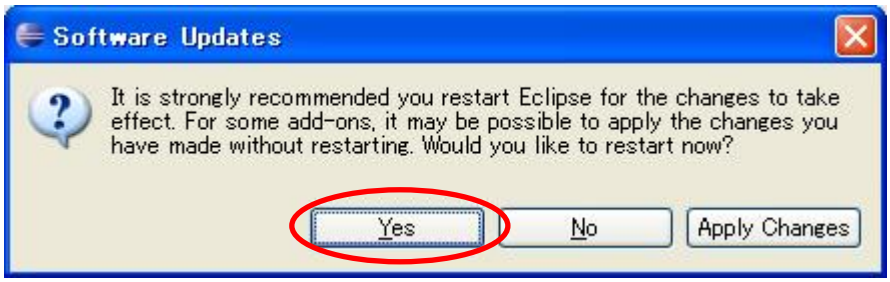
Install に"http://opensource.zylin.com/zylincdt "が追加されるのでチェックボックスにクリックしチェックを入れて Next ボタンを押す。




Install
Install Details
Review the items to be installed.
Name
Ver...
Id
Zylin Embedded CDT
4.8.1
com.zylin.cdt.feature.feature.group
Size: Unknown
Details
< Back
Next >
Finish
Cancel


Install
Review Licenses
Licenses must be reviewed and accepted before the software can be installed.
License text:
ECLIPSE FOUNDATION SOFTWARE USER AGREEMENT
March 17, 2005
Usage Of Content
THE ECLIPSE FOUNDATION MAKES AVAILABLE SOFTWARE, DOCUMENTATION, INFORMATION AND/OR
OTHER MATERIALS FOR OPEN SOURCE PROJECTS (COLLECTIVELY "CONTENT").
USE OF THE CONTENT IS GOVERNED BY THE TERMS AND CONDITIONS OF THIS
AGREEMENT AND/OR THE TERMS AND CONDITIONS OF LICENSE AGREEMENTS OR
NOTICES INDICATED OR REFERENCED BELOW. BY USING THE CONTENT, YOU
AGREE THAT YOUR USE OF THE CONTENT IS GOVERNED BY THIS AGREEMENT
AND/OR THE TERMS AND CONDITIONS OF ANY APPLICABLE LICENSE AGREEMENTS
OR NOTICES INDICATED OR REFERENCED BELOW. IF YOU DO NOT AGREE TO THE
TERMS AND CONDITIONS OF THIS AGREEMENT AND THE TERMS AND CONDITIONS
OF ANY APPLICABLE LICENSE AGREEMENTS OR NOTICES INDICATED OR REFERENCED
BELOW, THEN YOU MAY NOT USE THE CONTENT.
Applicable Licenses
Unless otherwise indicated, all Content made available by the Eclipse Foundation
is provided to you under the terms and conditions of the Eclipse Public
License Version 1.0 ("EPL"). A copy of the EPL is provided with this
I accept the terms of the license agreement
I do not accept the terms of the license agreement
< Back
Next >
Finish
Cancel

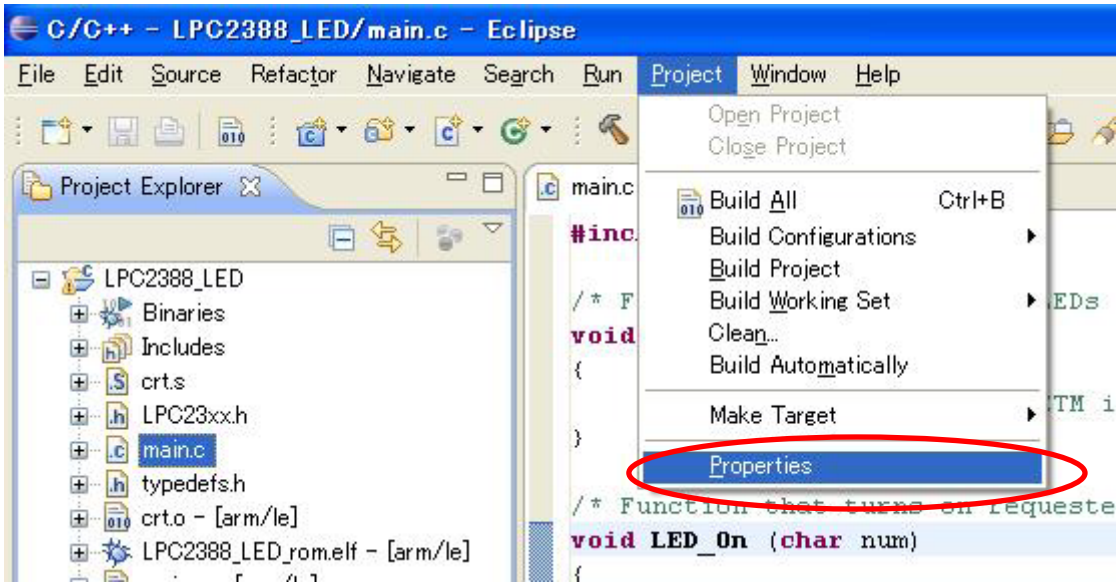

Install
Fetching com.zylin.embeddedcdt_4.8.1.jar .../plugins/com.zylin.embeddedcdt_4.8.1.jar
Always run in background
Install
Fetching com.zylin.embeddedcdt_4....s/com.zylin.embeddedcdt_4.8.1.jar
Run in Background
Cancel
<< Details



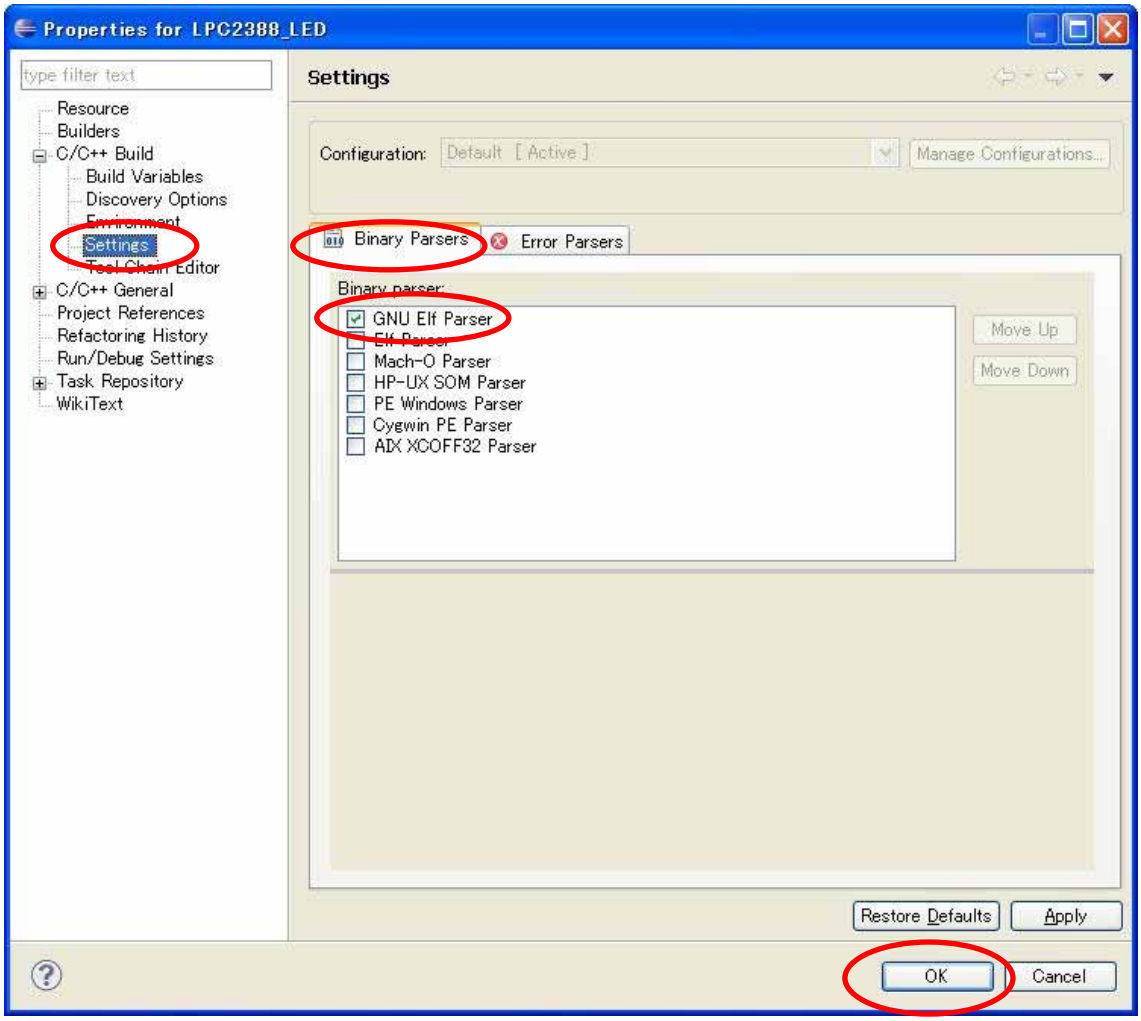
インストール完了したら、Yes ボタンを押して、Eclipse を再起動させます。

5.4 ビルドの設定

Eclipse の"Project"→"Preferences"を選択する

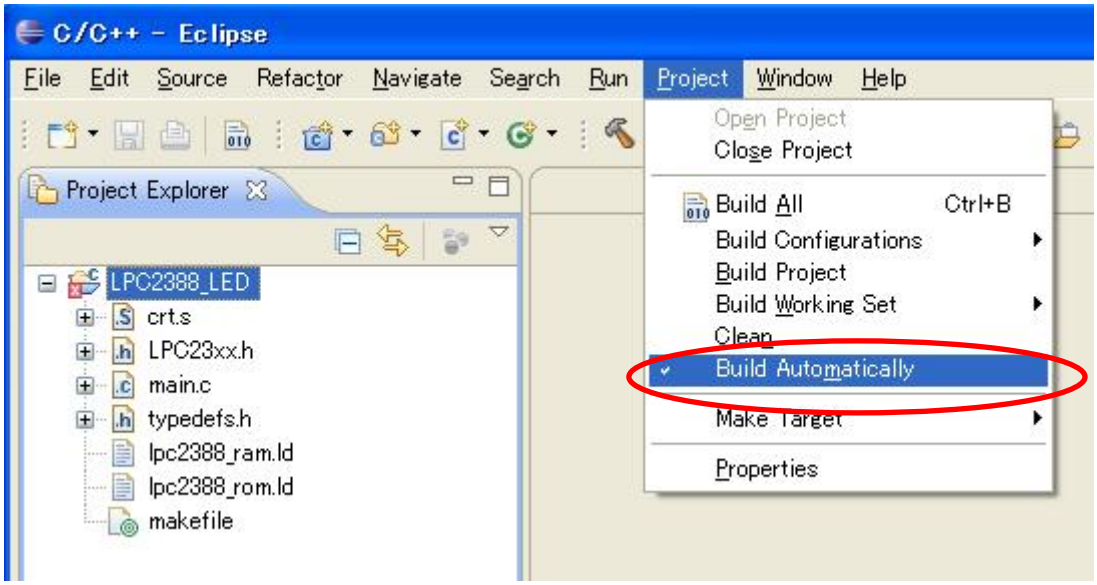


Preferences の"C/C++ Build"→"Settings"を選択し"Binary Parsers"タブの"GNU Elf Parser"にチェックを入れて OK ボタンを押します

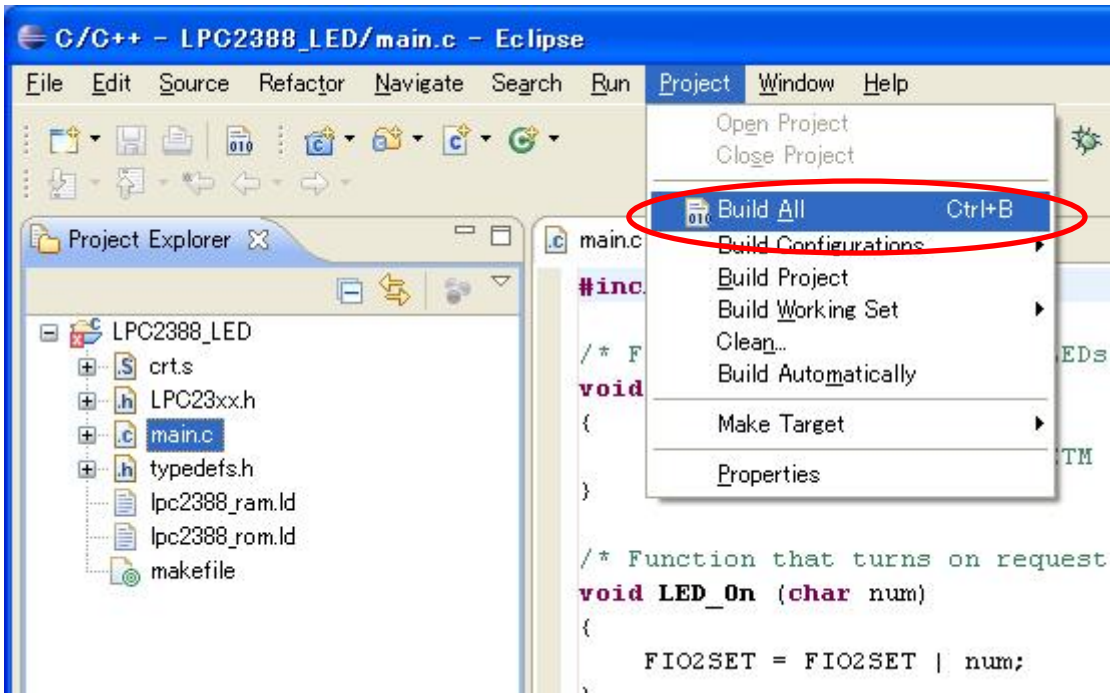


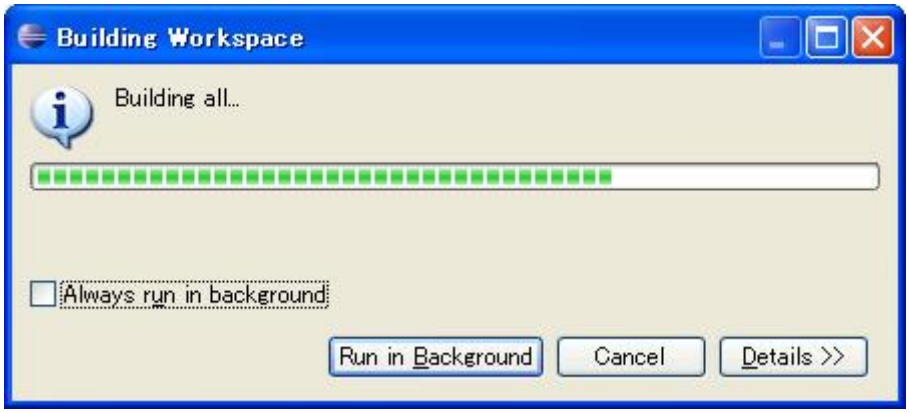
5.5 ビルド

Eclipse の"Project"→"Build Automatically"のチックを外してください。

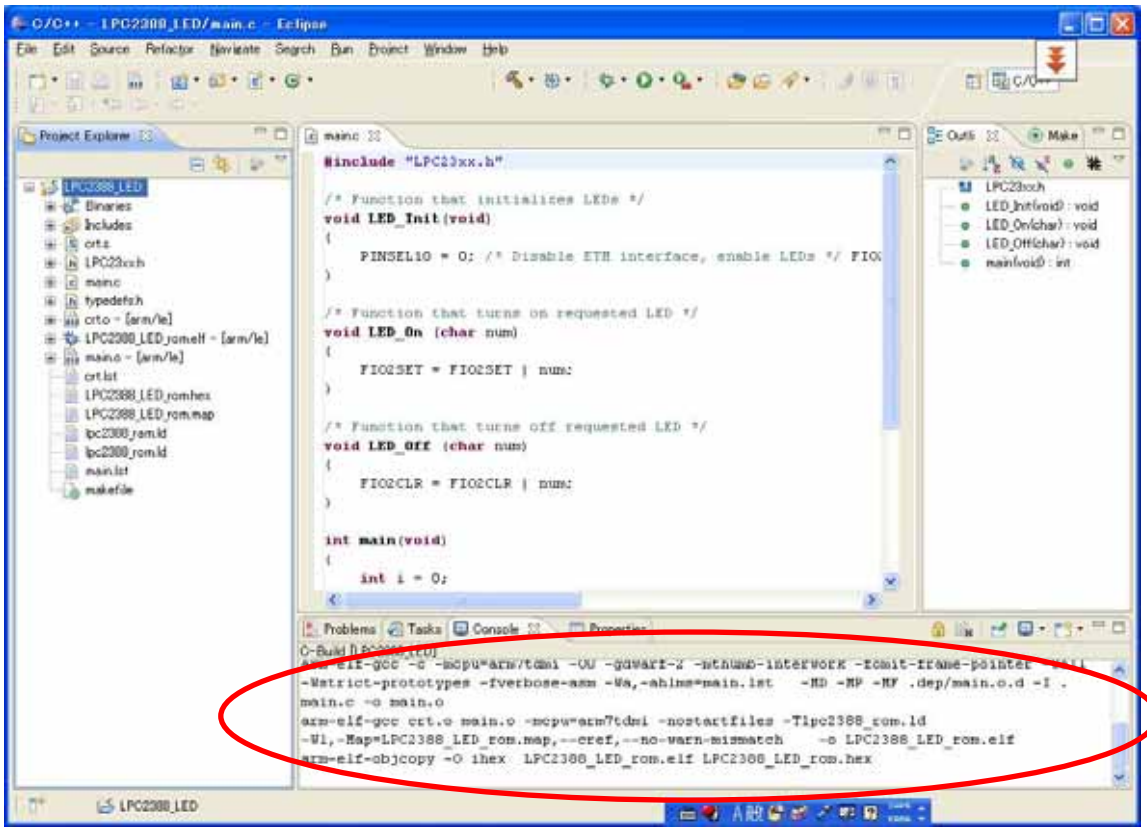


"Project"→"Build All"を選択するとビルドが行われます。





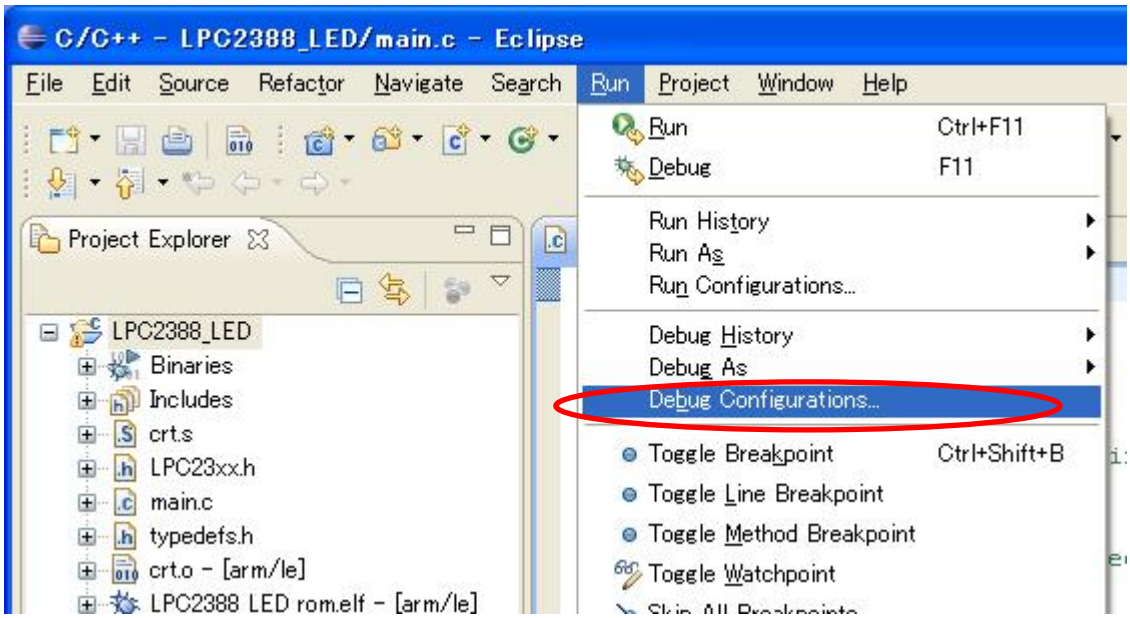
コンパイル中です。



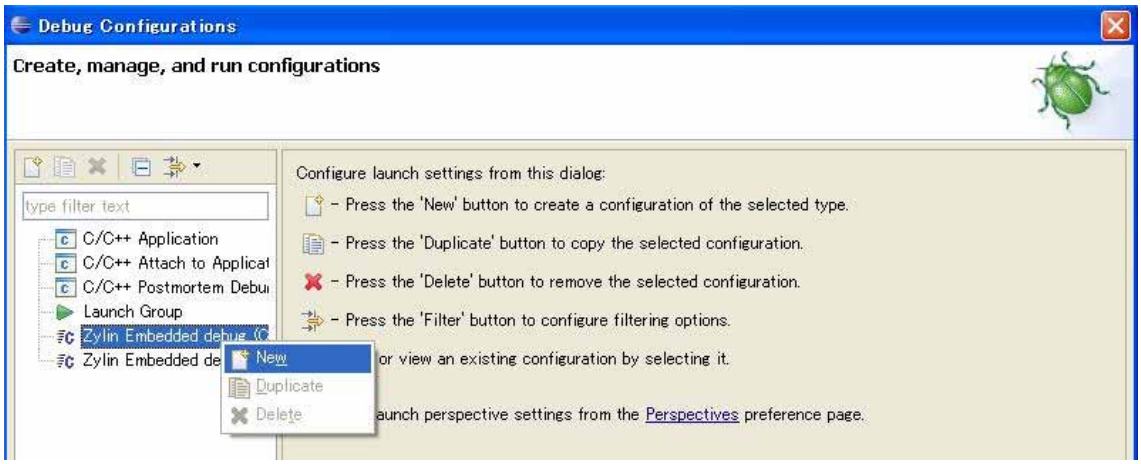
コンパイルが成功すれば、実行ファイル LPC2388_LED_rom.elf と LPC2388_LED_rom.hex を生成されます。

5.4 GDB の設定

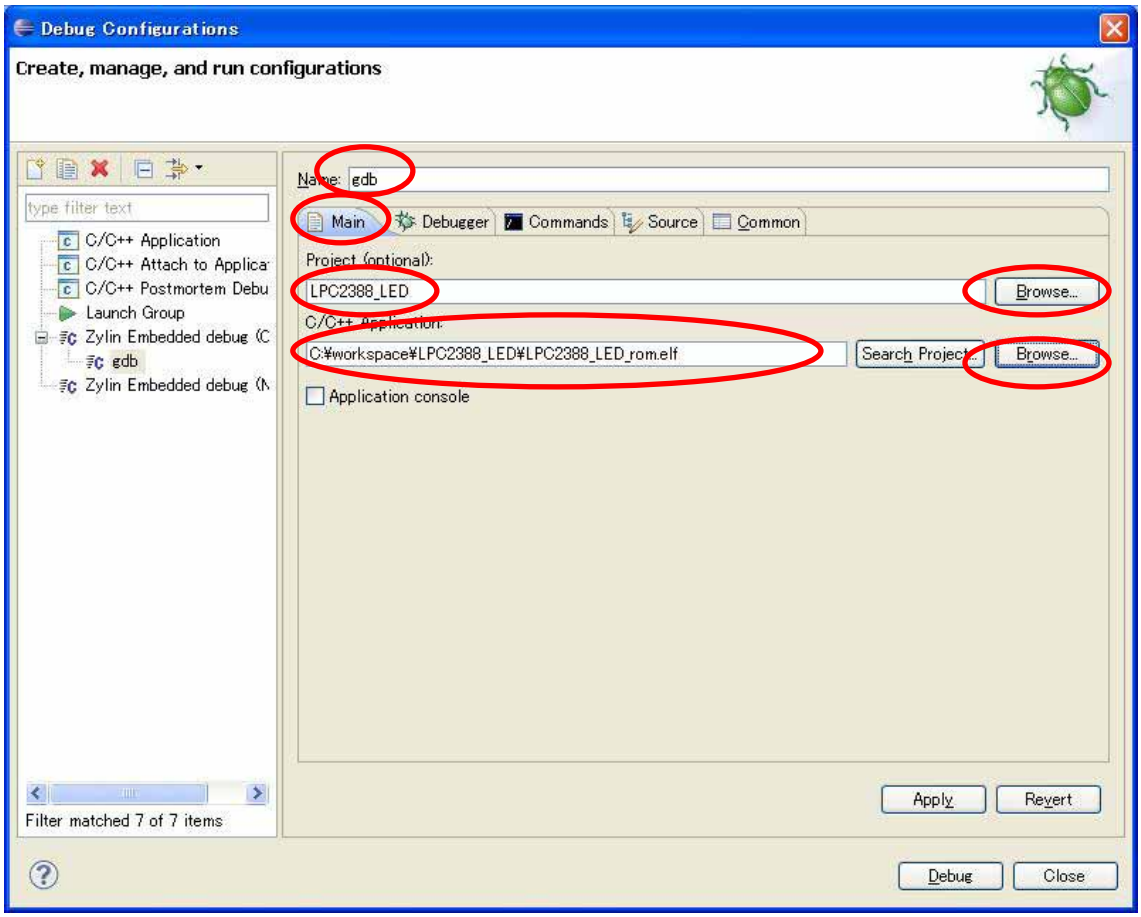
Eclipse の"Run"→"Debug Configurations..."を選択します。



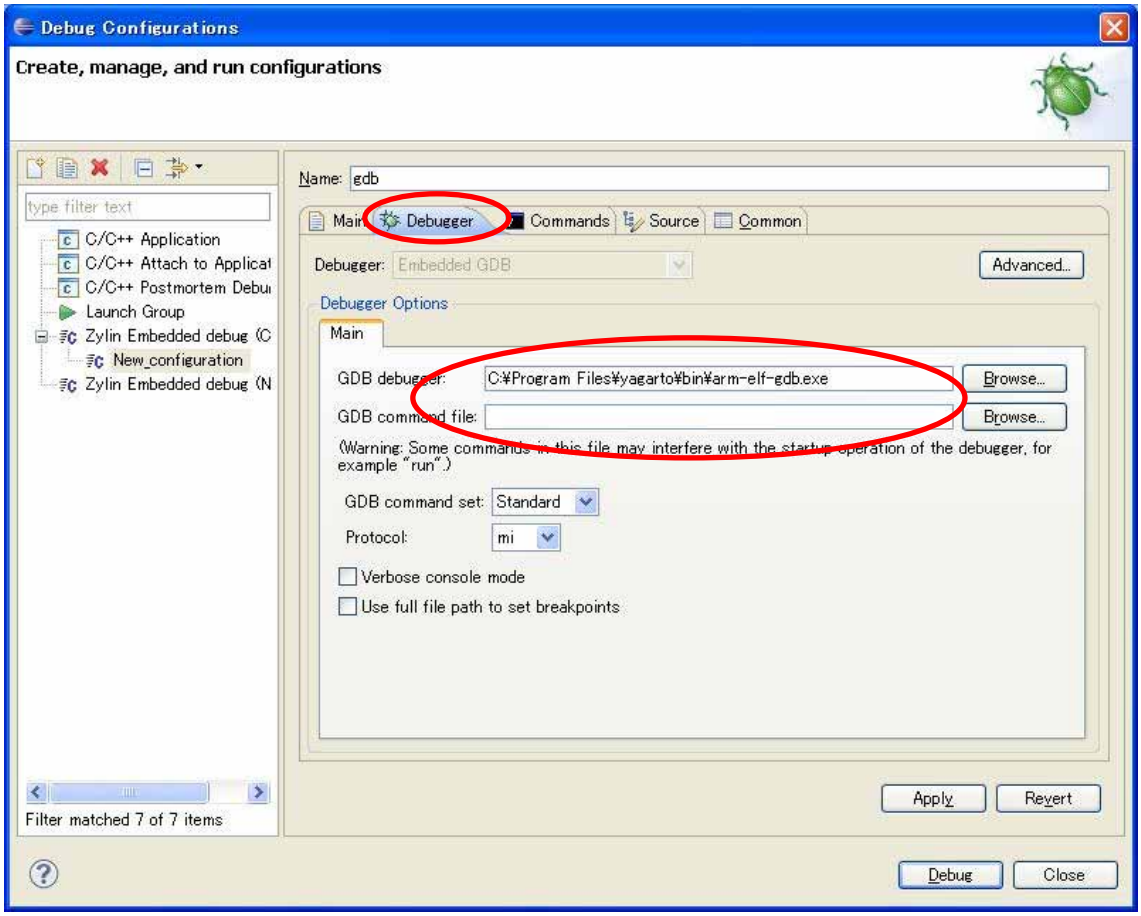
Debug Configurations の"Zylin Enbedded debug(Native)"を右クリックし"New"を選択する。



Name に適当な名前を入れる。例、"gdb"と入れます。Main タブの"Project"に"LPC2388_LED"、"C/C++ Appication:"に"C:¥workspace¥LPC2388_LED¥LPC2388_LED_rom.elf"と入力します。

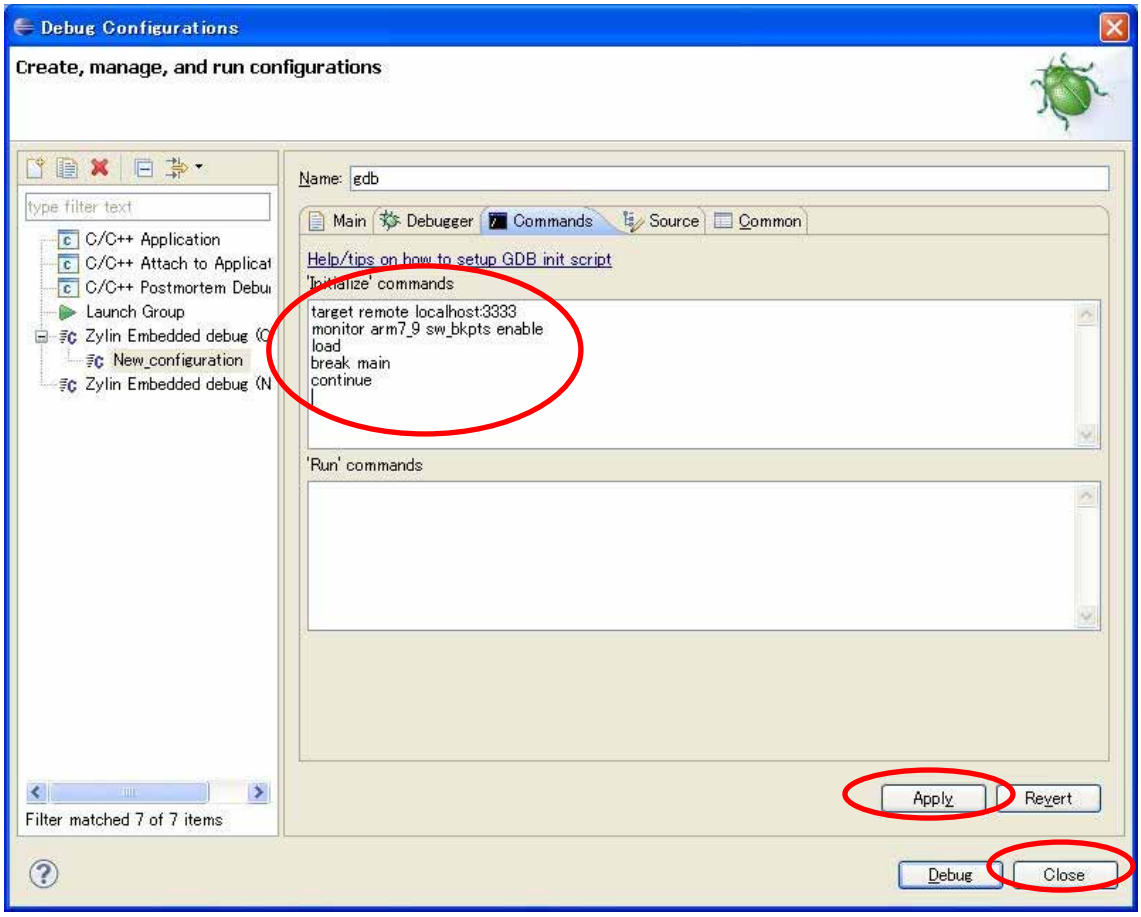


Debugger タブの"GDB debugger:"に"arm-elf-gdb"、"GDB command file:"に何も入力しません。



Commands タブの"'Initialize' commands"に下記の画面の様に入力します

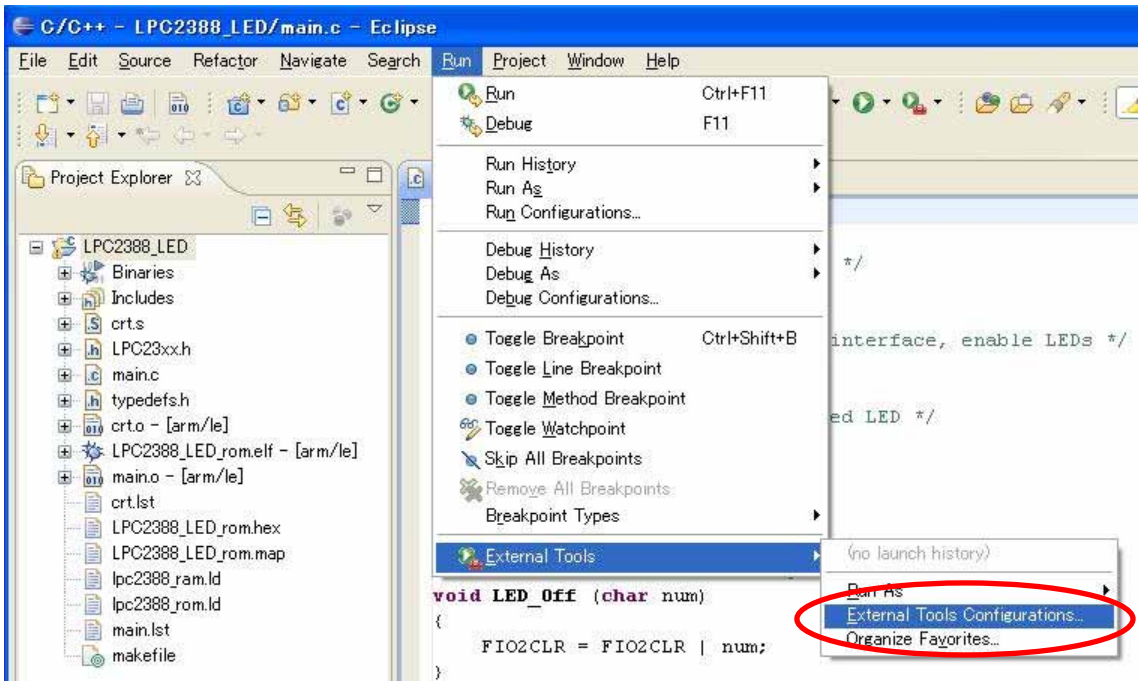
```
target remote localhost:3333
monitor halt
monitor step
load
break main
continue
```



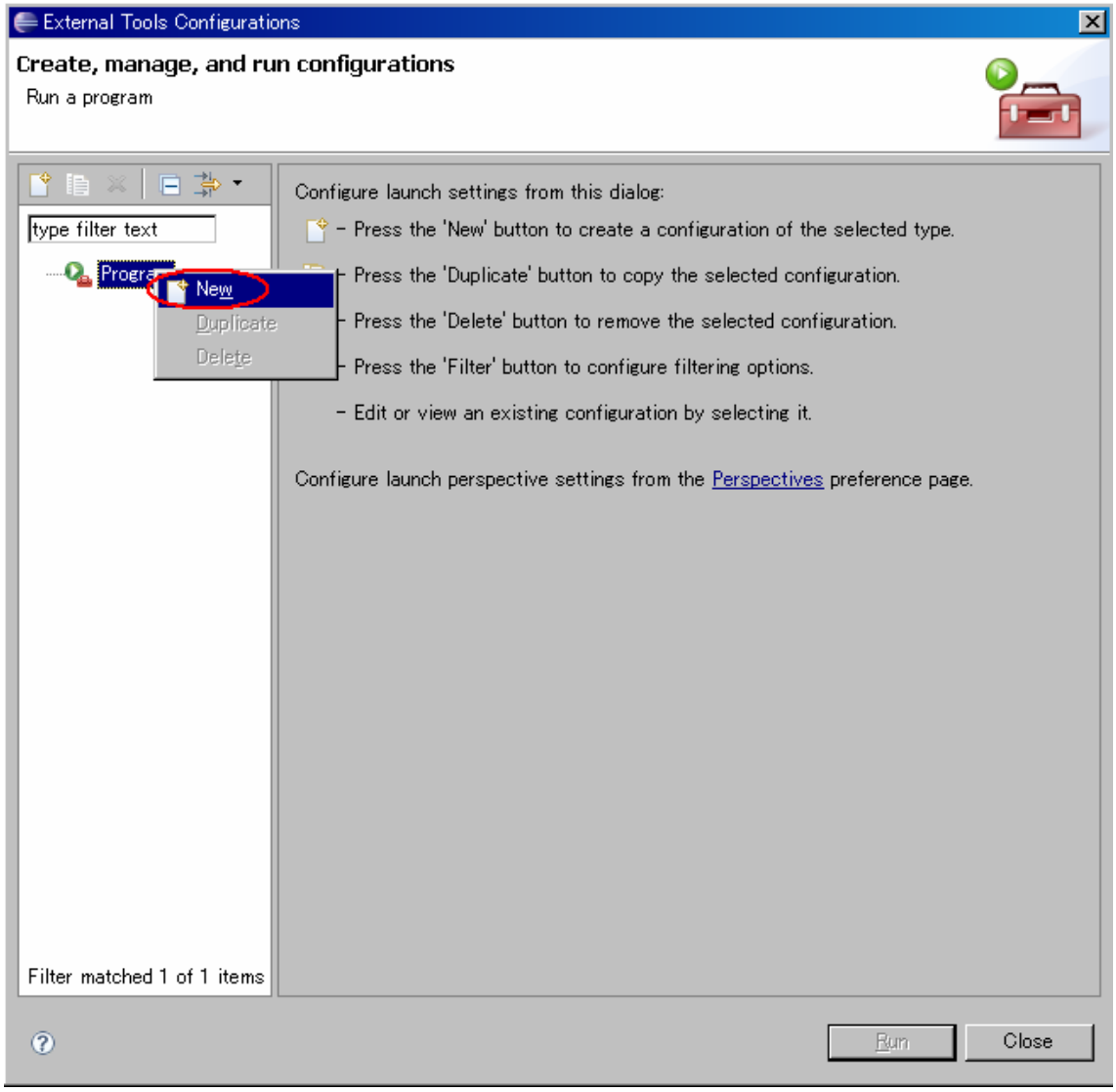
全てを入力し終えたら"Apply"ボタンを押し、"Close"ボタンを押します。

5.5 OpenOCD の設定

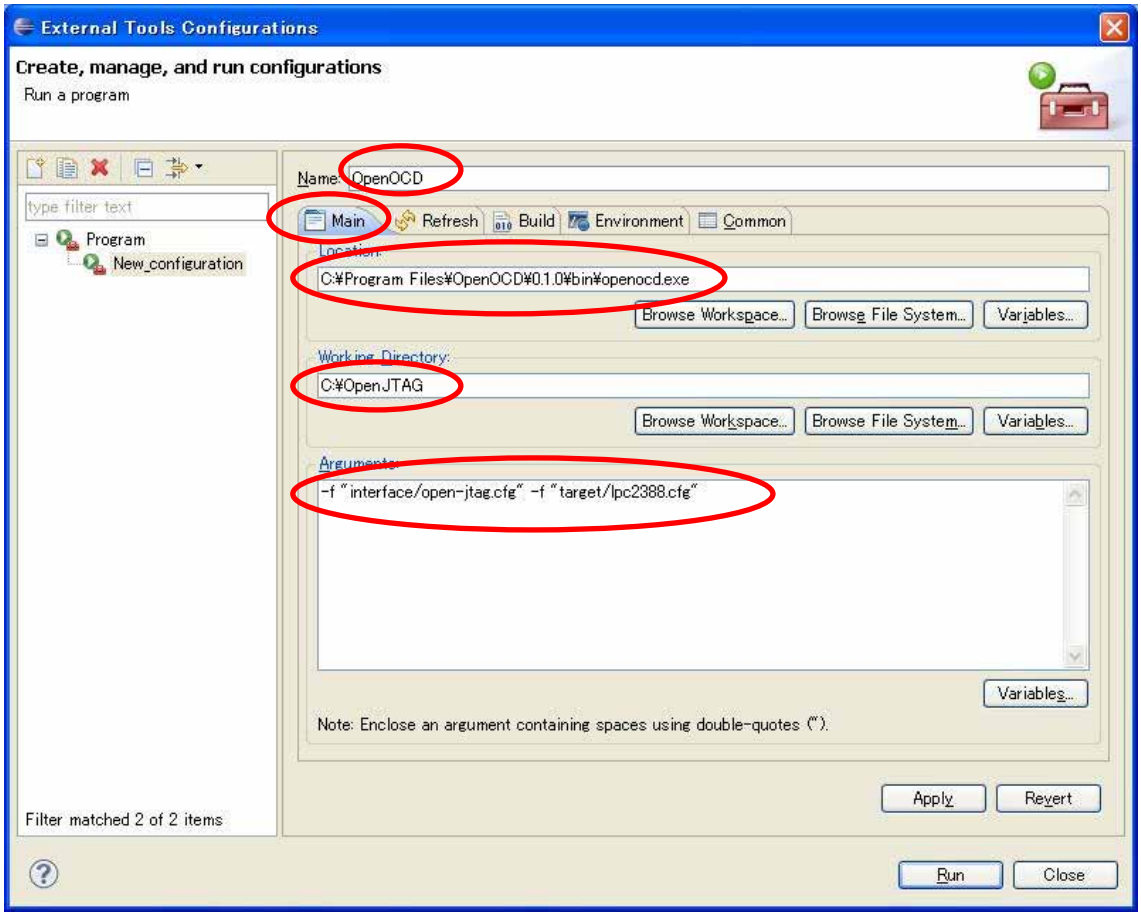
Eclipse の"Run"→"External Tools."→"External Tools Configrurations..."を選択します。

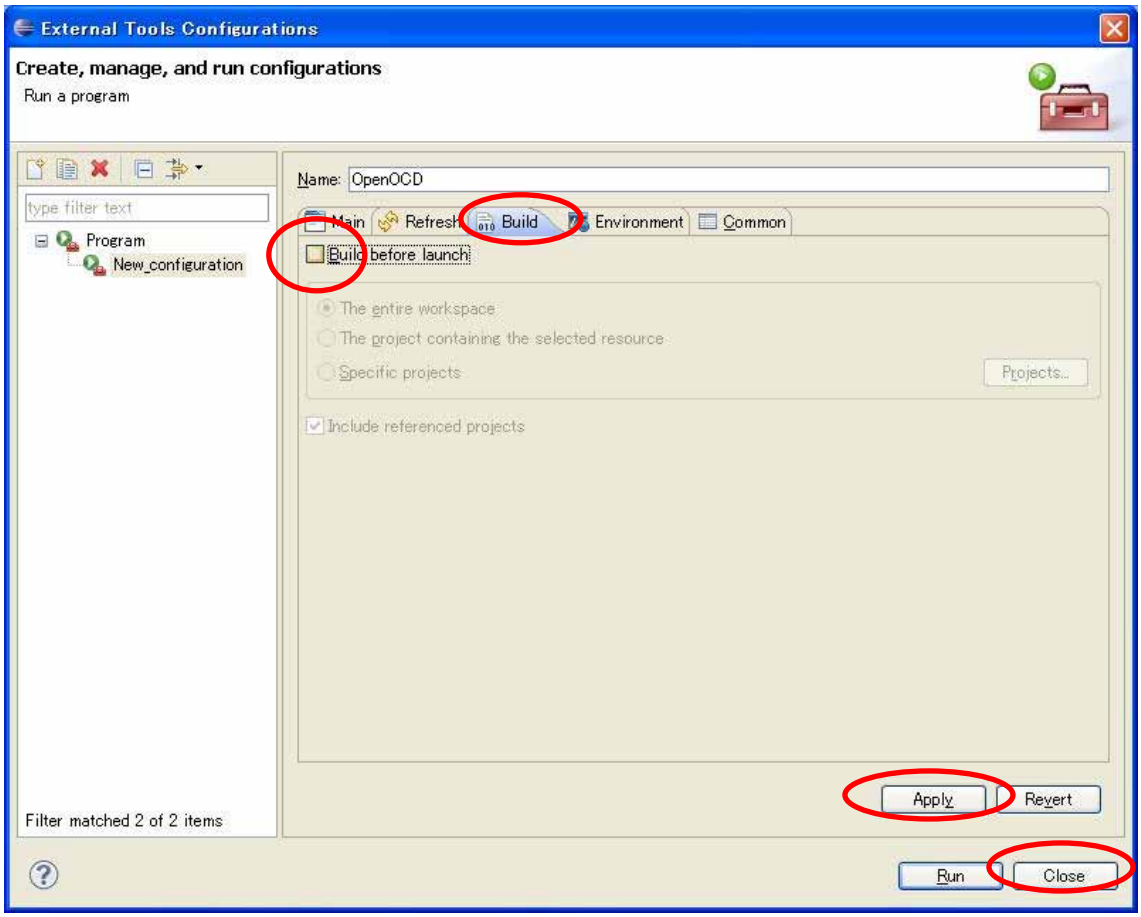


External Tools Configrurations の"Program"を右クリックし、"New"を選択します。



Main タブの"Name"に適当な名前を入力してください。私は ” OpenOCD"と入れました。
"Location:"に"C:¥Program Files¥OpenOCD¥0.1.0¥bin¥openocd"、
"Working Directory:"に"C:¥OpenJTAG"
"Arguments:"に-f "interface/open-jtag.cfg" -f "target/lpc2388.cfg"と入力します。





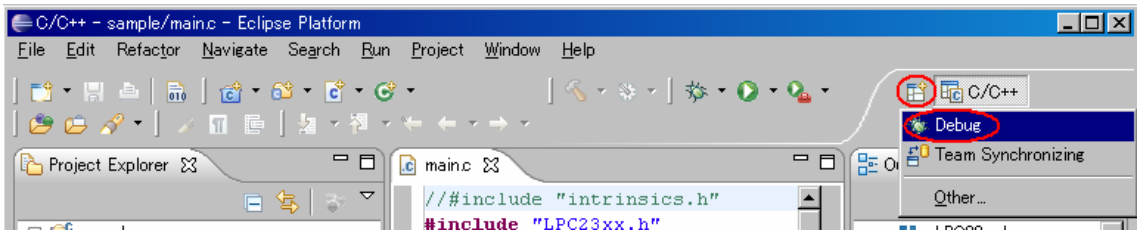
Build タブをクリックし"Build before launch"にチェックを外れます。全てを入力し終えたら"Apply"ボタンを押し、"Close"ボタンを押します。

5.6 デバッグ

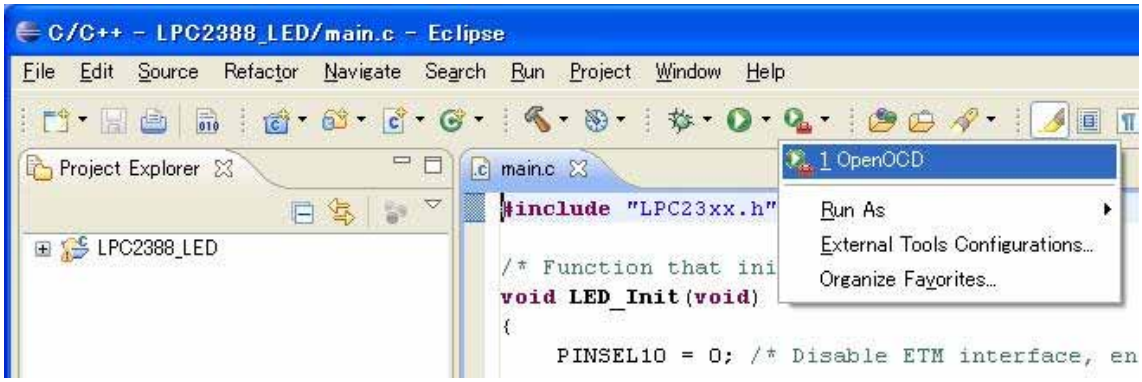
電源投入

1. OpenJTAG をターゲット(LPC2388 ボード)とパソコンに接続
2. ターゲットに電源を入れます

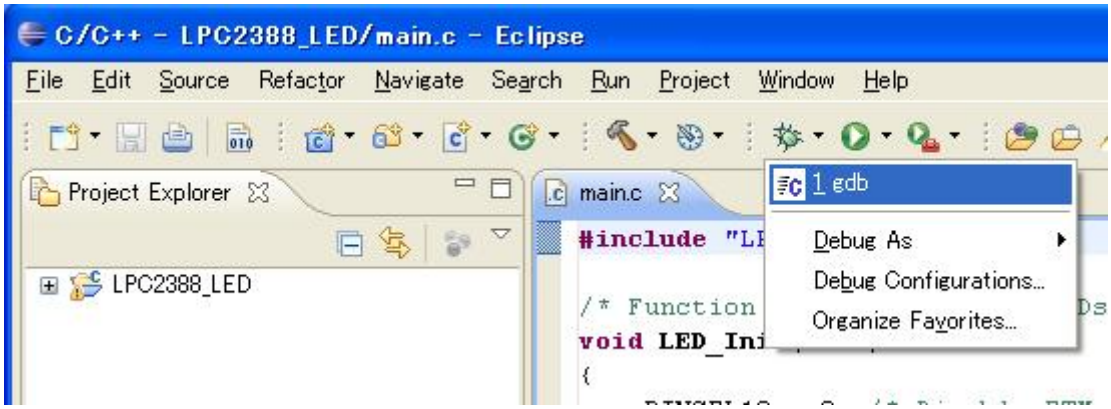
Eclipse 起動し、Eclipse の Open Perspective をクリックし Debug を選択

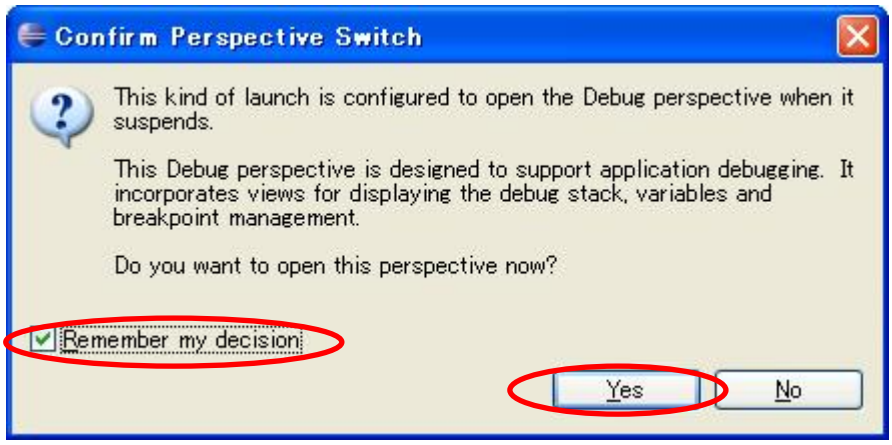


External Tools の▼ボタンをクリックし、OpenOCD を選択

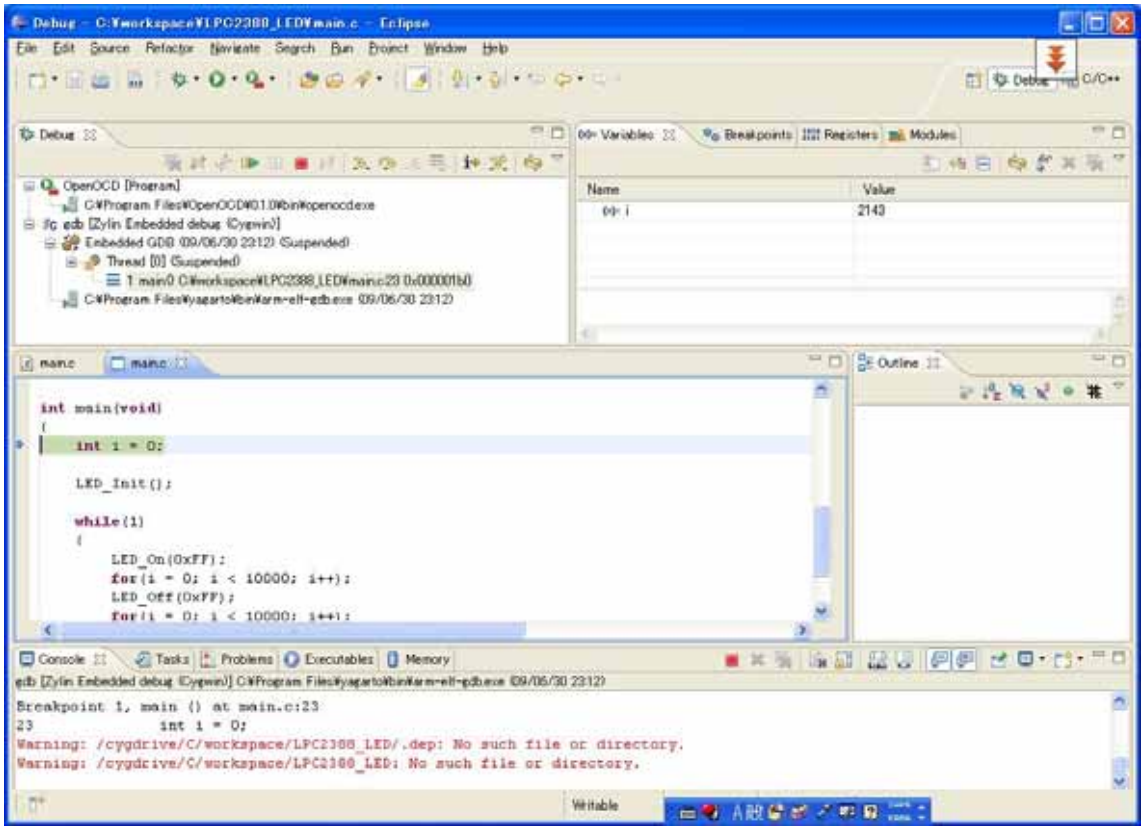


Debug の▼ボタンをクリックし、"gdb"を選択。





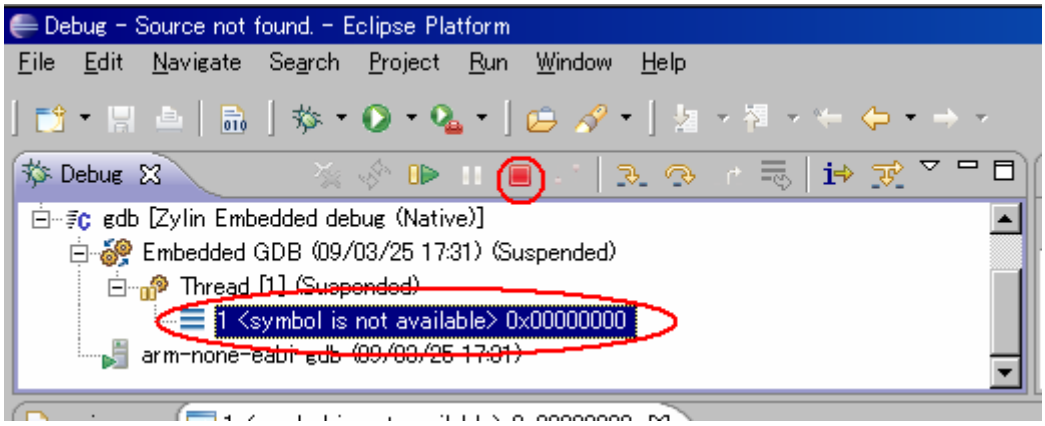
Yes ボタンを押して、デバッグが開始します。



5.7 デバッグ終了

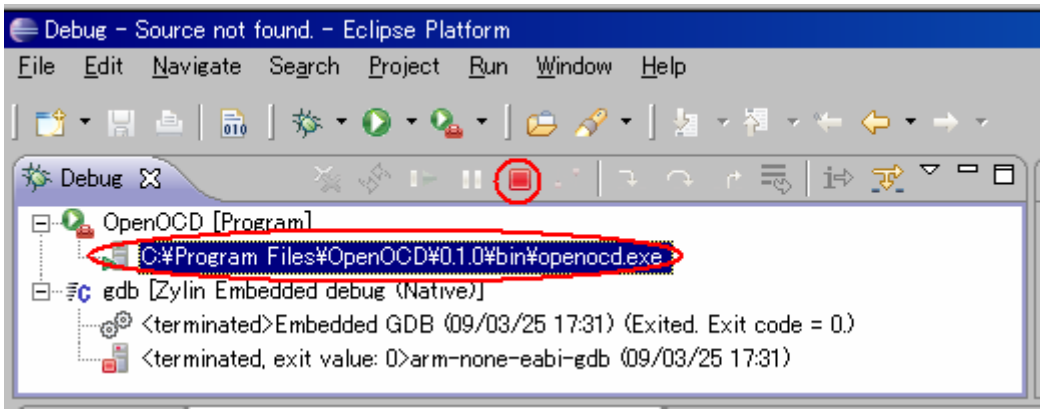
1）gdb の停止

Debug ウインドウの gdb の Thread を選択し、停止ボタンと押します



2）OpenOCD の停止

Debug ウインドウの OpenOCD の Thread を選択し、停止ボタンと押します



3）電源停止

ターゲットの電源を停止

4）OpenJTAG をターゲットから取り外す

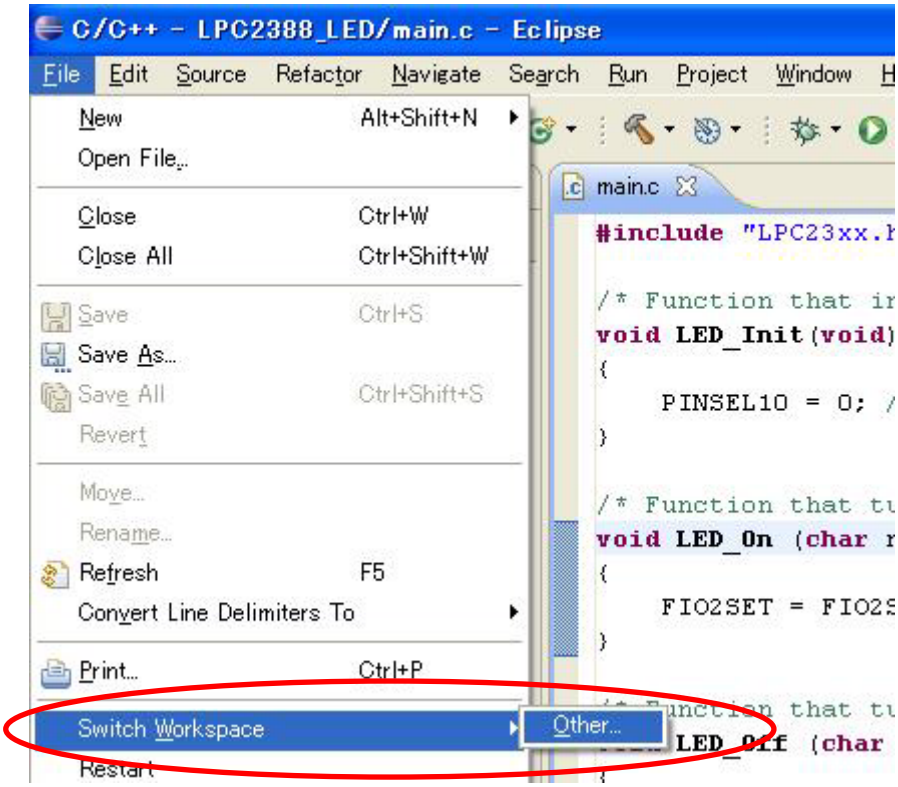
5）上記が面倒であれば Eclipse を終了しターゲットの電源停止、ARM-USB-TINY を取り外しでも OK です。

第六章 Web サーバサンプル

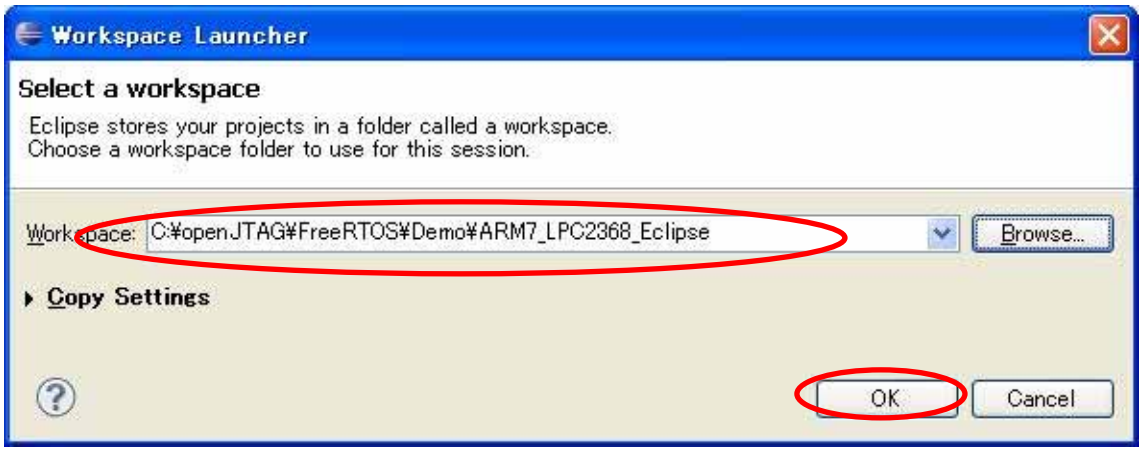
FreeRTOS フォルダは FreeRTOS というリアルタイム OS と uIP という TCP/IP スタックを利用して、LPC2388 用の Web サーバサンプルです。

6.1 プロジェクトを作る

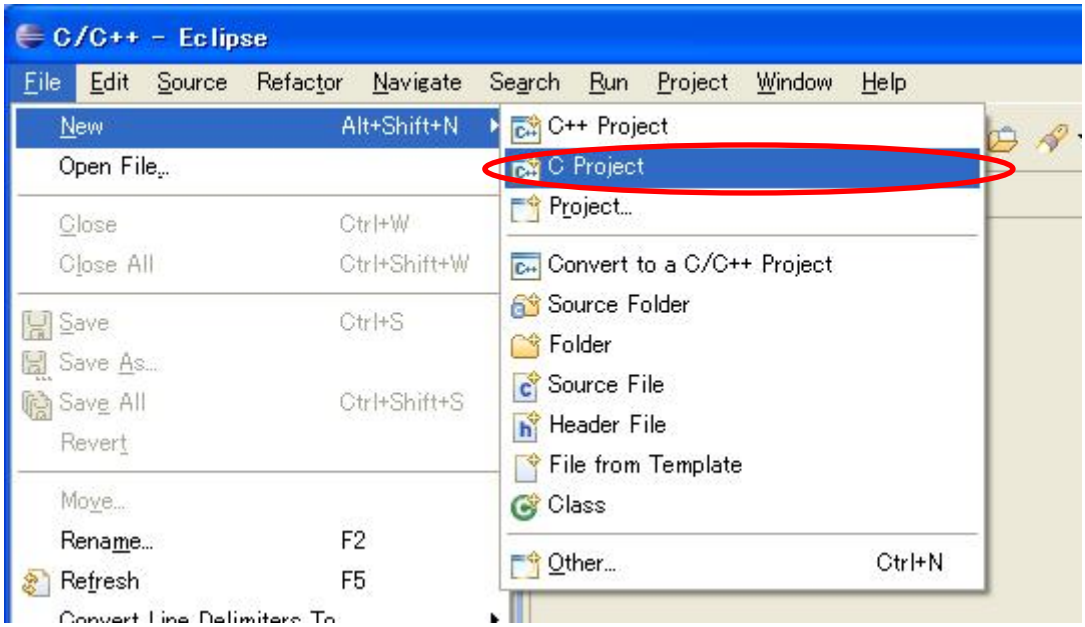
既存のフォルダを利用するため"File"→"Switch Workspace"を選択します。



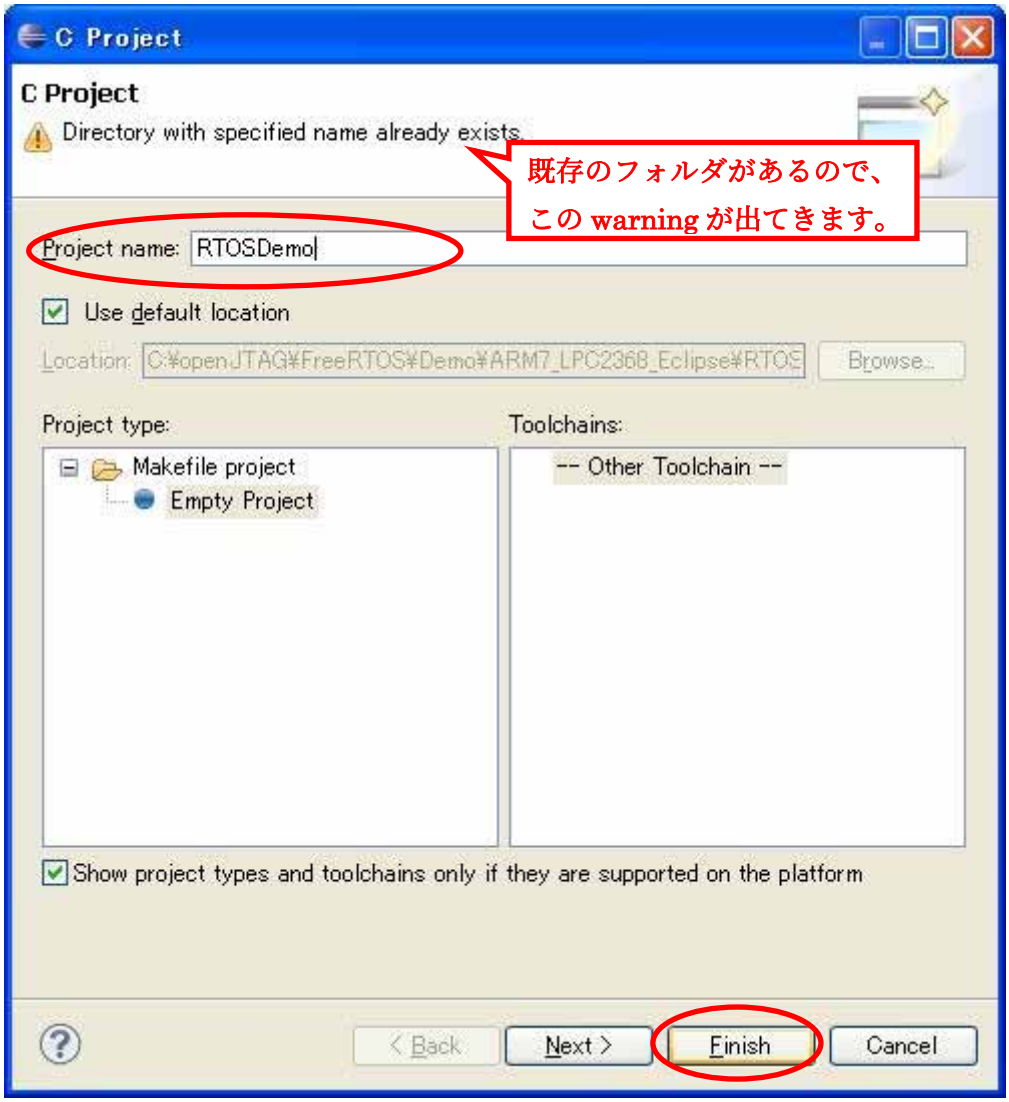
Workspace を" C:¥openJTAG¥FreeRTOS¥Demo¥ARM7_LPC2368_Eclipse "に変更しました。



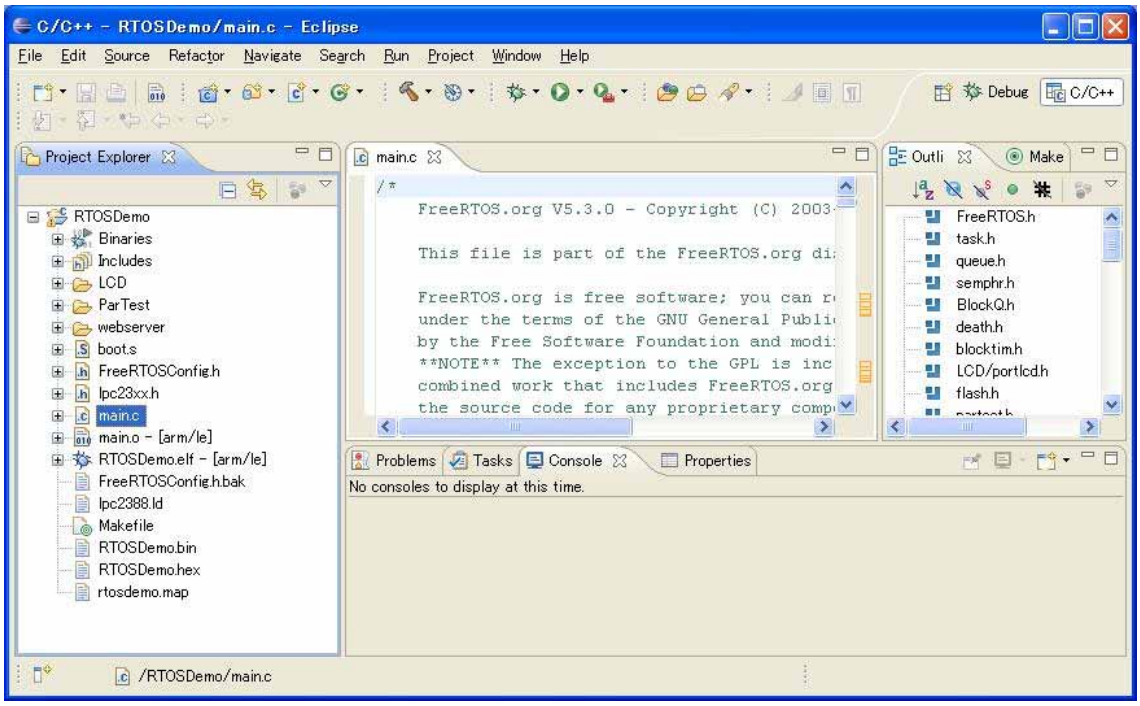
新規プロジェクトを作成するため"File"→"New"→"C Project"を選択します



プロジェクト名を聞かれるので RTOSDemo を入力し Finish ボタンを押します。



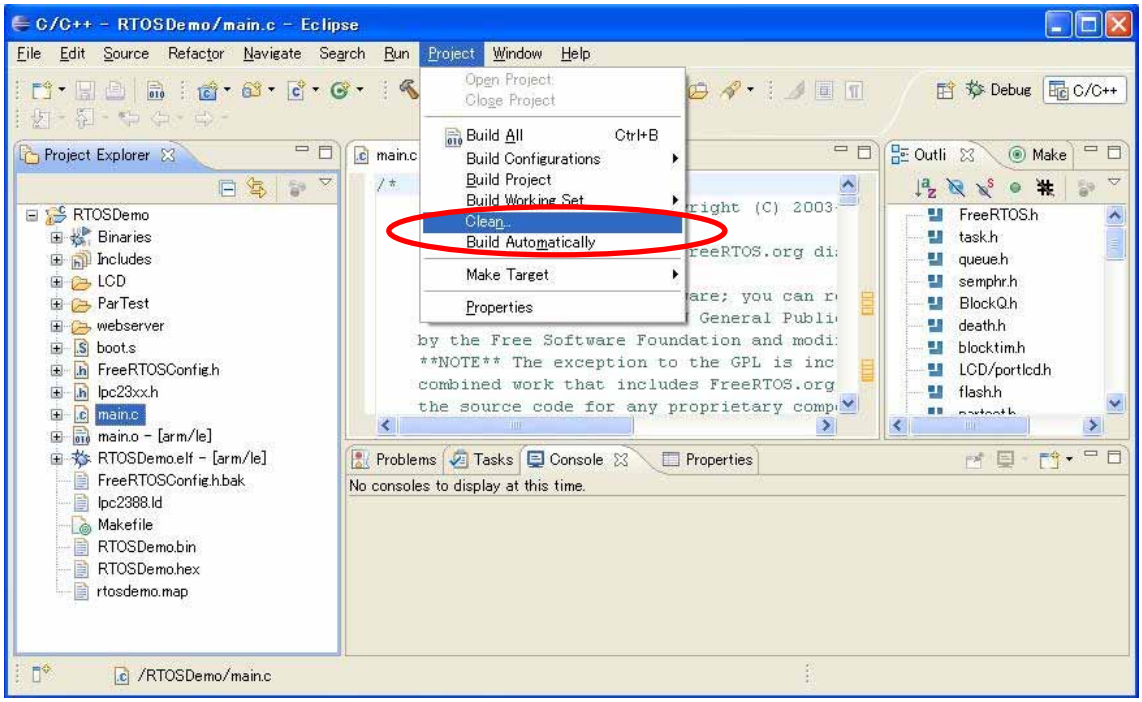
Project Explorer の"RTOSDemo"プロジェクトの左にある＋をクリックするとファイルの一覧が表示されます。

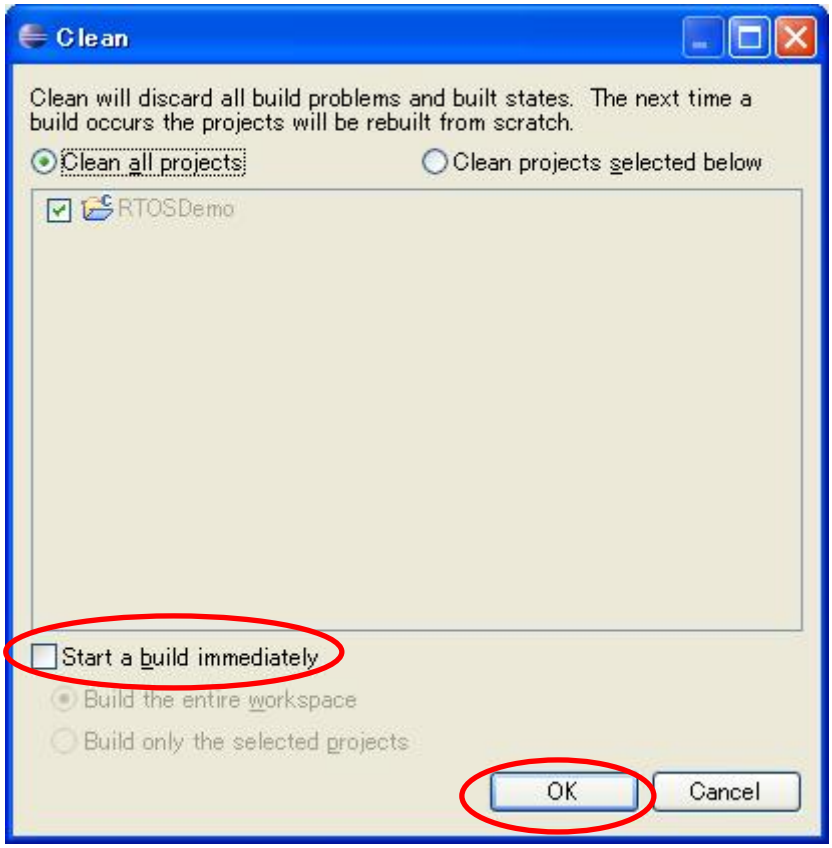


第五章と同じ手順で Eclipse を設定してください。

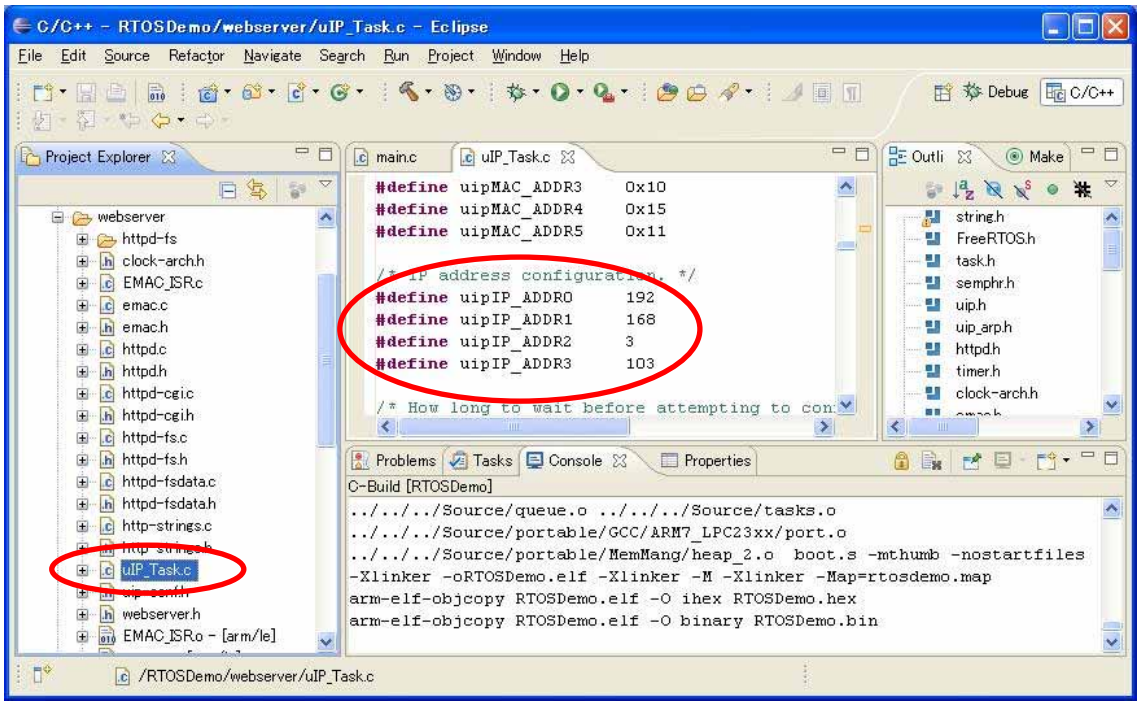
6.2 ビルド

コンパイル済みなので、再コンパイルする前に、クリアしてください。"Project"→"Clean"を選択します。

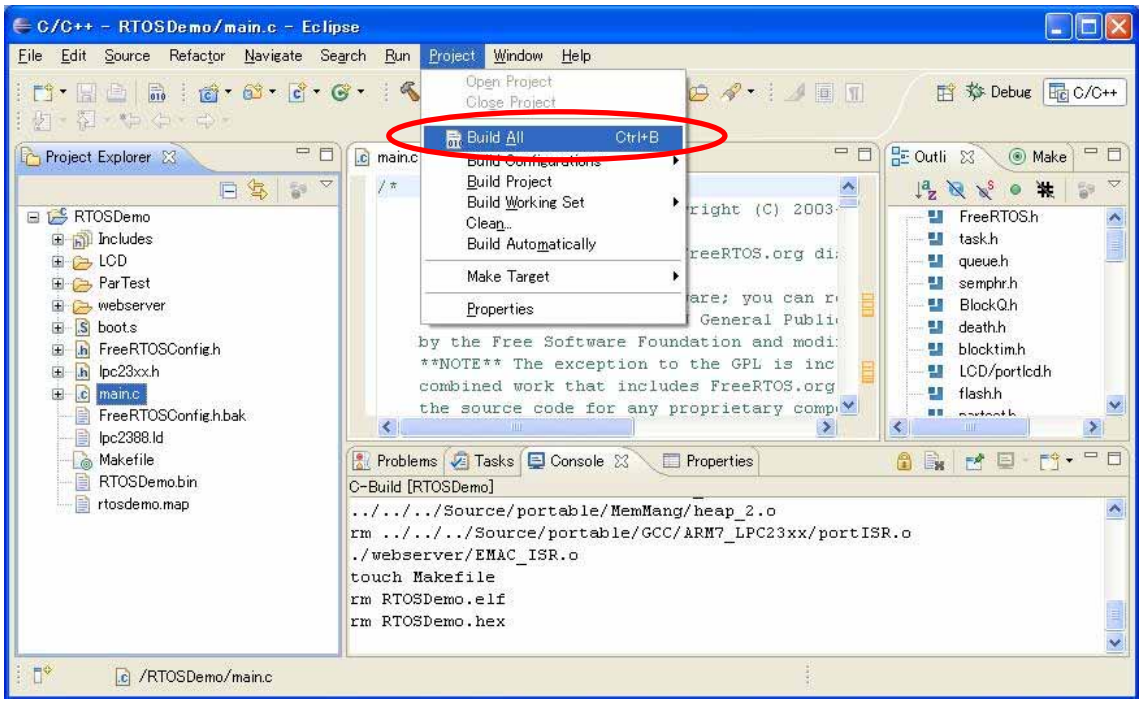




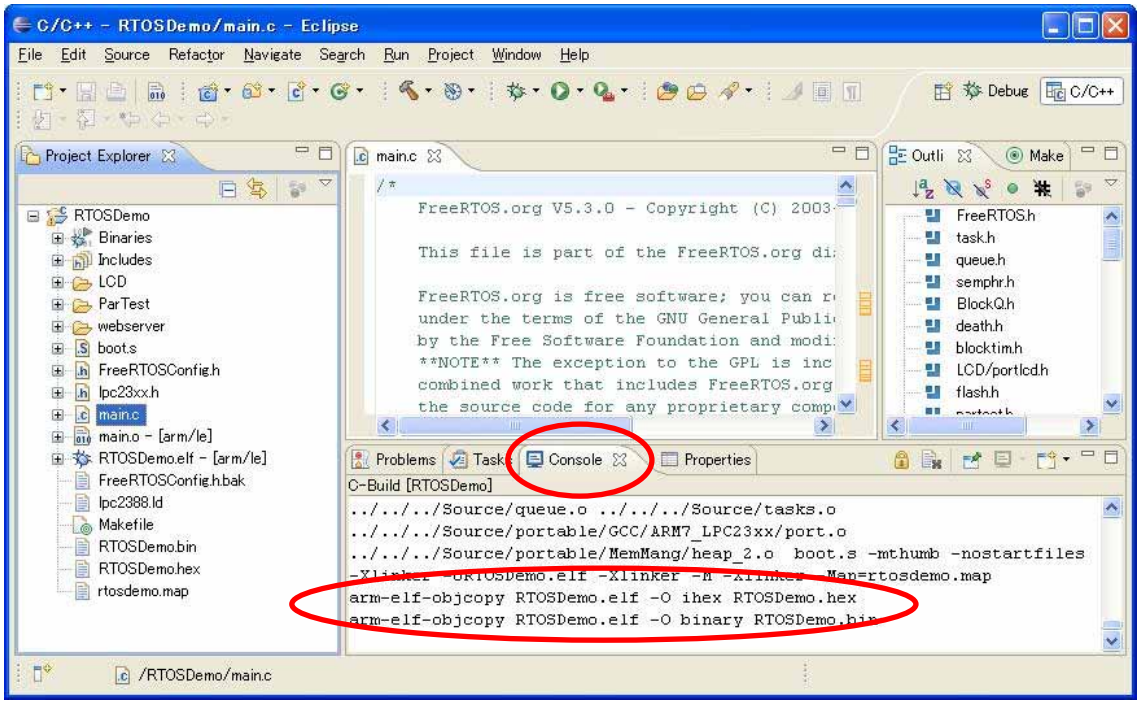
再コンパイルする前に、自分のネットワーク環境によって、ターゲットの IP アドレス(webserver/uIP_Task.c)を変更してください。



"Project"→"Build All"を選択するとビルドが行われます。



コンパイルが成功すれば、実行ファイル RTOSDemo.elf と RTOSDemo.hex を生成されます。

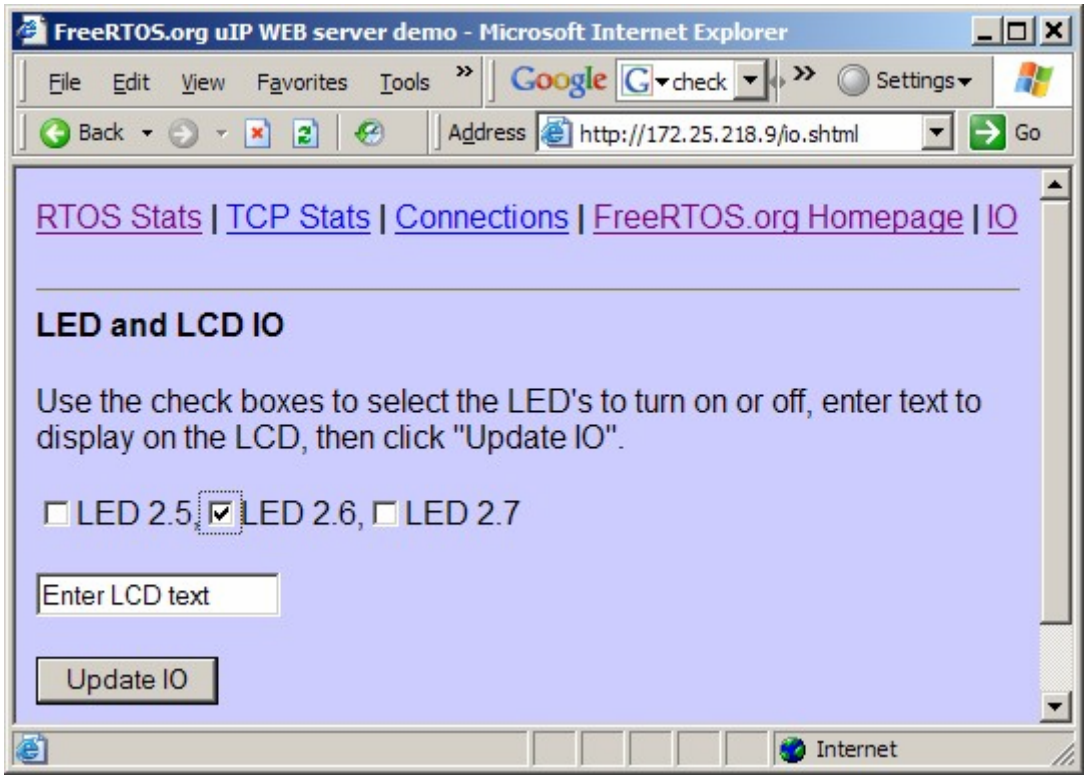


6.3 書き込みと実行

OpenOCD の書き込みスクリプト：

```
arm7_9 dcc_downloads enable
wait_halt
sleep 10
poll
flash probe 0
#flash protect 0 0 26 'off'
flash erase 0 0 26
flash write 0 ./RTOSDemo/RTOSDemo.bin 0x0
reset run
sleep 10
shutdown
```

LPC2388 ボードで実行する様子：



詳しい情報はこちらにご参照ください。

http://www.freertos.org/index.html?http://www.freertos.org/portlpc2368_Eclipse.html